Fast Iterative Regularisation Methods
(Alternative Format Thesis)

Sabaté Landman, Malena

Award date:
2021

Awarding institution:
University of Bath

Fast Iterative Regularisation Methods
(Alternate Format Thesis)

Sabaté Landman, Malena

Award date:
2021

Awarding institution:
University of Bath

University Of Bath
Department of Mathematical Sciences

EPSRC Centre for Doctoral Training in
Statistical Applied Mathematics (SAMBa)

"Fast Iterative Regularisation Methods"

Doctoral thesis

submitted by
Malena Sabaté Landman
for the degree of Doctor of Philosophy

Lead supervisor
Dr. Silvia Gazzola
Other supervisors
Prof. Melina Freitag
Prof. Manuchehr Soleimani

February 2021

Declaration of any previous submission of the work

The material presented here for examination for the award of a higher degree by research has not been incorporated into a submission for another degree.

Declaration of authorship

I am the author of this thesis, and the work described therein was carried out by myself personally, with the exception of Chapters 2 and 4, which contain research articles that originated from collaboration with my supervisor Silvia Gazzola, and Chapter 3 which contain research articles that originated from collaboration with Silvia Gazzola and James Nagy.

Acknowledgments

It feels like time has released its pressure for a moment as I finish this thesis. In my pause I see another milestone walking past me, and before the path takes me onto the next stage, I want to stop and thank some of the many beautiful things I have encountered along the way.

I want to give thanks to my research community, for giving me a space to nestle, and to my life teachers for showing me how to ask questions. Most importantly, to Silvia, for being my guide but also my companion through this storm. Because we did not always take the easy route, but she always made sure there was a way through, and she never left me alone. Because she believed in my success, because she is selfless and brilliant, and because working with her is a pleasure and it is fun. To Jim, for showing me that research can be as peaceful as a fresh fall eve in Atlanta. Because he showed that being optimistic is allowed as long as it comes with hard work. For his invaluable support and hospitality, and for teaching me how to step into research without feeling it is a borrowed space. To my examiners, Tony and Rosie, for making the term defending a thesis obsolete, and for lending me their eyes to see my work from another perspective. Finally, to all the teachers who paved my way before this PhD; to Alejandra for shipping me out to Albion, to Juan Carlos for making me a mathematician by mistake, and to the maestra Claudia for gifting me with the present of curiosity.

For the years I spent in Bath, I want to thank the_studio for being my home and my ship, to my unconditional team; and to Myowny for keeping me safe in her ark. I want to thank Andy for keeping my glass of spirituality half full, and to thank Jen for being honest and bold: for having the courage to face me with truths and the heart to accept me with kindness. I want to thank John for opening my eyes to my biases, for letting me be family, for coming without conditions and, occasionally, with a cup of tea. I want to thank my dear friend Emiko for speaking through food and doraibus, for always being ears, for listening out for the full version of myself; and for knitting a world full of Mofs and Wobs and cheese to wrap me up on the so-so days. I also want to thank Alastair, Naomi and Logan for setting an extra plate on the table on the days when, either for joy or for sorrow, I needed a home.

I am grateful for the department, for the echoes of the friendships forged on its corridors that slip through zoom calls during the current pandemic. I, too, reach my hands through the internet trying to hold on to those connections: to the coffees with Ben and Matt, to Elizabeth (just being Elizabeth), to Hayley's freshness, to Aoibheann's care (thanks for the proof-reading!) and to Lizzi's understanding smile. I am grateful to all my friends around the world, and for the visits I received since I moved to the UK; from Mi, Jordina, Pat, Cris, Alex, Marc, Iván, Ignasi, Rocco, Pol and Sandra. Thanks for understanding my gibberish and my metaphors, and for always being there.

I want to thank Tito for the ritual of reading the news every morning, and with whom I would have liked to share my graduation. I want to thank my family for supporting me in the distance and hiding (badly) how much they would have liked that Bath was at walking distance from Barcelona. Like my mum before me, I am an emigrant now; but I left my hometown with my mother's voice, my dad's hands, and my sister's courage for the days

I miss the land where I was born. This new country I landed in is a choice I am grateful for. I want to thank her for offering fertile soil for my longing roots, and for this beautiful language that makes me dream in new colours, and to whom I present an ‘s’, as a little homage, in the title of this thesis.

Contents

Abstract

This thesis assembles three published papers containing original research in the area of regularization techniques for large-scale linear discrete inverse problems.

In order to present the natural framework of this thesis, a general introduction to large-scale linear discrete inverse problems is given first, along with a brief description of the nature of these problems that motivates the need for regularization. The review on background material, presented in Chapter 1, has not been written with the aim of being exhaustive, but to provide a common setting for the work in Chapter 2, Chapter 3 and Chapter 4. In fact, most of the content in Chapter 1 is presented in the corresponding introductions of the papers included in Chapter 2 [GL20], Chapter 3 [GNL21] and Chapter 4 [GL19], but it is presented in a slower pace, with illustrative examples and more detailed explanations. In particular, Chapter 1 focuses on the description of Krylov subspace methods and different variational regularization techniques that are suited for recovering different desired properties of the inverse problem solution. Moreover, Chapter 1 describes two canonical examples of linear imaging problems: image deblurring and computed tomography. The rest of the thesis follows naturally from this general introduction, as Chapter 2, Chapter 3 and Chapter 4 describe methods that combine Krylov subspace methods with three different variational regularization schemes, and that are tested on imaging problems. All chapters are rich in numerical examples to illustrate the performance of the different regularization strategies studied in this thesis.

Chapter 2 includes the survey paper [GL20] on regularizing Krylov projection methods and hybrid methods, which combine the use of Tikhonov regularization and Krylov subspace methods. Moreover, [GL20] addresses the difficulty of finding a good regularization parameter for Tikhonov regularization, and presents a novel scheme based on bilevel optimization that interlaces iterations of standard parameter choice strategies with Krylov iterations. In particular, when the discrepancy principle is used as the parameter choice rule, [GL20] provides a theoretical proof of convergence of the solution computed using the novel scheme to the Tikhonov solution with regularization parameter determined by the discrepancy principle.

Chapter 3 includes paper [GNL21], that presents two new algorithms (IRW-FGMRES and IRW-FLSQR) to compute sparse solutions of large-scale linear discrete inverse problems. The new approach is based on finding a solution of a smooth approximation of the ℓ_2-ℓ_p regularization problem, for $0 < p \leq 1$. This is achieved by constructing a sequence of quadratic tangent majorants to a smooth approximation of the ℓ_2-ℓ_p problem, for $0 < p \leq 1$, that are partially solved using flexible Krylov methods. Moreover, [GNL21] provides a theoretical proof of convergence of the solutions computed using IRW-FGMRES and IRW-FLSQR to the solution of the smooth approximation of the ℓ_2-ℓ_p problem.

Chapter 4 includes paper [GL19], that presents a new algorithm, TV-FGMRES, to compute solutions to large-scale linear discrete inverse problems using total variation (TV)

regularization. In the context of imaging problems, this scheme allows a good edge-recovery in the approximated solution. Similarly to the strategy described in Chapter 3, the non-smooth functional associated to TV regularization is approximated in this chapter by a sequence of quadratic functionals. This leads to a sequence of quadratic minimization problems that are partially solved using flexible Krylov methods and efficient implementation strategies.

Finally, Chapter 5 gives some last remarks and conclusions, as well as future possible directions of work.

Chapter 1

Background on regularization for large-scale linear discrete inverse problems

Inverse problems involve the reconstruction of a hidden object from some possibly noisy measured data after it has been transformed by some forward process, so that the object is just measured indirectly. In this thesis we will assume that the forward process is linear and known. In particular, we will focus on discrete inverse problems arising from suitable discretizations of Fredholm integral equations of the first kind, which arise in a variety of scientific and engineering applications. We are interested in imaging problems and, more specifically, in image deblurring problems (e.g, like those arising in astronomical and biological applications) and computed tomography (e.g, used in medicine and also in different applications in industry); for more details see [Han10, Vog02, HNO06, HH93, EHN96] and the references therein. The problems we are interested in have two main features that make them interesting yet challenging to solve. First, linear inverse problems tend to be ill-posed, as the reconstructed solution is very sensitive to perturbations in the measurements. Therefore, in order to obtain meaningful approximations to the original solution, one must resource to regularization, i.e., replace the original problem with a closely related problem that has better stability properties. Second, real-world applications of inverse problems are often large-scale, resulting in very computationally demanding tasks when the matrices involved do not have an exploitable structure. In this case, the only feasible approach to recover a solution to the ill-posed large-scale linear inverse problem is to apply an iterative solver.

In Section 1.1 an overview of linear inverse problems and their properties is given, with the aim of describing their nature and motivating the need for regularization. Section 1.2 introduces two specific imaging problems, namely: image deblurring and computed tomography, and describes common test problems in this setting. Section 1.4 introduces different variational regularization schemes, where the regularized solution is computed solving an optimization problem. Finally, Section 1.5 introduces iterative regularization methods and dwells on classical and flexible Krylov methods for linear problems.

1.1 Linear inverse ill-posed problems

Consider a general Fredholm integral equation of the first kind

$$\int_{\Omega} k(s,t)f(t)dt = g(s), \tag{1.1}$$

where $s, t \in \Omega \subset \mathbb{R}^q$, Ω is compact and Jordan measurable with positive measure and $k \in L^2(\Omega \times \Omega)$ is a known kernel that is associated to the underlying forward model (so that the integral operator is compact [EHN96, Chapter 2]). Integral equations have a smoothing effect on the measurements: components of $f(t)$ with higher frequency will appear damped in $g(s)$ compared to components with lower frequency [Han10]. This can be understood under the light of the Riemann-Lebesgue lemma, that states that if $f(t) = \sin(2\pi pt)$ and p tends to infinity, then $g(s)$ in (1.1) goes to zero for any kernel $k(s,t)$ with sufficient regularity [Han10]. Note that the smoothness of $k(s,t)$ is related to the severity of the dampening. For this reason, in the context of the inverse problems, where $g(s)$ is known and $f(t)$ is the desired solution, the amplification in $f(t)$ of high frequency terms in $g(s)$ is more severe when $k(s,t)$ is smoother. In particular, noise components of small amplitude but high frequency that might appear in $g(s)$ will create large oscillations in $f(t)$.

Suitable discretizations of equation (1.1), e.g., quadrature (or collocation) methods and expansion (or projection) methods [Han10, Section 3.1], lead to linear discrete inverse problems of the form

$$Ax = b, \quad \text{where} \quad A \in \mathbb{R}^{m\times n}, \quad b \in \mathbb{R}^m. \tag{1.2}$$

Here, x corresponds to a discretization of $f(t)$ in the domain Ω. A is the matrix representation of the integral operator with the kernel $k(s,t)$, which is usually of ill-determined rank (i.e., the singular values of A decay and cluster at zero without an evident gap between two consecutive ones). Finally, b is the vector of measurements corresponding to a discretization of $g(s)$ (which is usually corrupted by some added noise e, so that $b = b_{true} + e$). Note that, when analyzing linear discrete inverse problems of the form (1.2), we implicitly assume that the discretization of the integral operator is sufficiently fine so as to, loosely speaking, "inherit" the behaviour of the original continuous problem.

Although, for some inverse problems, A may be unknown, in this thesis we will always assume both b and A are known, and we want to recover a good approximation of x. The noise vector e is not known in (1.2) but for the algorithms presented in this thesis we will assume that e can be well approximated by Gaussian white noise, i.e., the entries of the vector e are uncorrelated and are sampled from a Gaussian distribution with zero mean and the same variance. Note that this can be extended to the case of non-white Gaussian noise, i.e., with covariance matrix $C_e \neq I$, using a procedure commonly referred to as "whitening". More details will be given in Section 1.4.

As it will be mentioned in the following sections and chapters, the choice of algorithms that are appropriate to solve equation (1.2) is conditioned by the normality of the coefficient matrix A, which for A square is defined as

$$A^T A = AA^T.$$

Note that non-square matrices cannot be normal, and that the coefficient matrix of the normal equations associated to problem (1.2), i.e., $A^T A$, is always normal.

1.2 Imaging Problems

In this section two imaging problems that can be expressed in the general form (1.1) will be introduced. In particular, the aim is to give a more detailed description of the main examples used to test the performance of the new algorithms presented in the different papers included in this thesis [GL19, GL20, GNL21]. These descriptions are not included in the publications due to space constraints.

Although both problems will be introduced in the continuous framework dictated by equation (1.1), in practice we work with the discretized equation (1.2). In particular, square gray-scale digital images can be represented by two-dimensional arrays $X \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$, where $\sqrt{n}$ is a scalar, whose entries capture the intensity of the signal at a given section of the image, or *pixel*. Note that, for simplicity of the notation, we assume that all images are square, but the results can be trivially generalized to non-square images. Although the values of $[X]_{i,j}$ are usually quantized and stored as unsigned integers between 0 and 255, we will convert them to double precision floating point in our computations. Typically, images of either 256×256 or 512×512 pixels are used in the examples included in this thesis. Note that, to obtain a linear system of the kind (1.2), a vectorized version of the image, $x \in \mathbb{R}^n$, is used, which can be obtained by stacking the columns of X so that

$$x_i = [X]_{k,l} \quad \text{for} \quad i = (l-1)\sqrt{n} + k, \quad l, k = 1, ..., \sqrt{n}. \tag{1.3}$$

Most of the test problems used in this thesis, as well as some of the general solvers used to validate the presented algorithms, can be found in the MATLAB packages IR Tools [GHN18] and Restore Tools [NPP].

For all the applications considered in this thesis, we will assume that the matrix A in (1.2) is known (at least in the form of actions of A and sometimes A^T on vectors). In the synthetic examples used in this thesis, we will also consider the exact solution x_{exact} to be known, so that the reconstruction error can be monitored for different solvers. Although the exact right-hand side b_{exact} could be easily obtained as $b_{exact} = Ax_{exact}$, this situation is unrealistic. Indeed, the assumption that the model used to compute the reconstruction x is the same model that has been used to generate the measurements b, is usually known as "inverse crime" [Han10, Chapter 7.2]. To avoid this situation, and generate a more realistic problem, some test data are generated with a slightly different A (for example, deblurring examples in IR Tools [GHN18] might employ different boundary conditions for the forward and the inverse problems). Finally, the noise corrupting the measurements is considered to have a noise level $nl = \|e\|_2/\|b_{true}\|_2$, or, equivalently, a signal to noise ratio $SNR = 20\log_{10}(\|b_{true}\|_2/\|e\|_2)$; and the entries of e in (1.2) are drawn from uncorrelated Gaussian distributions with zero mean and equal variances, i.e., Gaussian white noise.

1.2.1 Image Deblurring

Image deblurring consists of reconstructing digital images that have suffered from a blurring process. This process often occurs inside the camera, usually because the camera lens is out of focus, or because the object or the camera are moving while the picture is being taken. There are also other situations in which we can find blur, for example in astronomical imaging, where it can be caused by turbulence in the atmosphere. In this framework, we will consider 2D problems where the blurring is a known process that is either linear or can be well approximated by a linear operator.

Image deblurring problems can be formulated in the general form (1.1), where the kernel $k(s,t)$ is a function that models how a single point in the image is distorted in the measurements (i.e., how it is spread across its neighborhood), and it is usually referred to as point spread function (PSF). When dealing with spatially-invariant blur, i.e., $k(s,t) = k(s-t)$, equation (1.1) reduces to a convolution. A couple of schematic examples of an image deblurring problem of this kind can be found in Figures 1.1 and 1.2 .

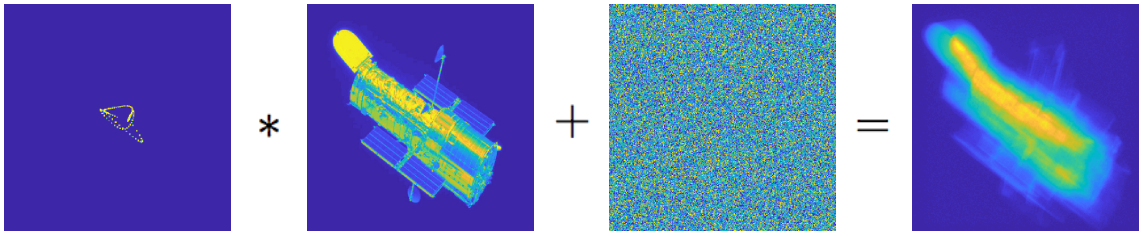


Figure 1.1: Sketch of an image deblurring problem with 'shake' blur. We can observe a representation of a PSF function being convoluted to the *satellite* image [GHN18], with added white noise, to obtain a blurred noisy version of the image.

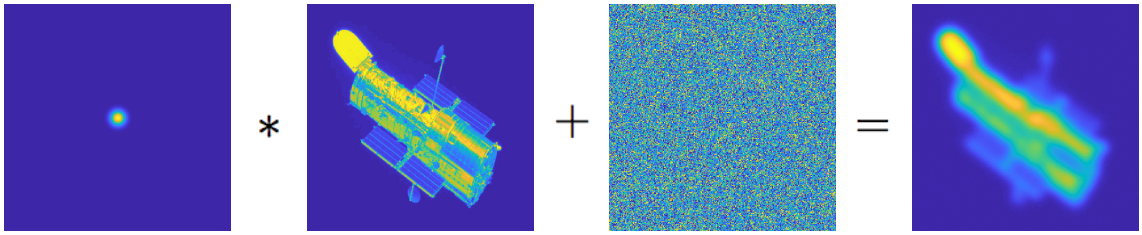


Figure 1.2: Sketch of an image deblurring problem with Gaussian blur. We can observe a representation of a PSF function being convoluted to the *satellite* image [GHN18], with added white noise, to obtain a blurred noisy version of the image.

In the following, we explain how the coefficient matrix A in (1.2) can be constructed in a way that provides a good intuition on the structure of the matrix. This approach is given in [HNO06, Chapter 3].

A discrete point source is an array $E_k \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$ where all entries but one (with pixel value of 1), are zero. In vectorized form, this corresponds to the canonical vector $e_k \in \mathbb{R}^n$ for $1 \leq k \leq n$, where the point source is located at the pixel of E_k corresponding to the kth component of e_k. By definition, the discrete measurement of a point source after blurring is given by the discrete PSF centered around the point source location; see Figure 1.3. If we denote by $P_k \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$ the PSF matrix corresponding to the blurred version of the point source E_k, and by $p_k \in \mathbb{R}^n$ its vectorized version, then the columns of the coefficient matrix A can be constructed as

$$a_k = Ae_k = p_k,$$

where a_k denotes the kth column of A. For example, when A represents a two dimensional Gaussian blur (e.g., atmospheric blur) centered around the pixel of E_k corresponding to the kth component of the vectorized image e_k, then the PSF matrix $P_k \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$ is defined component-wise as

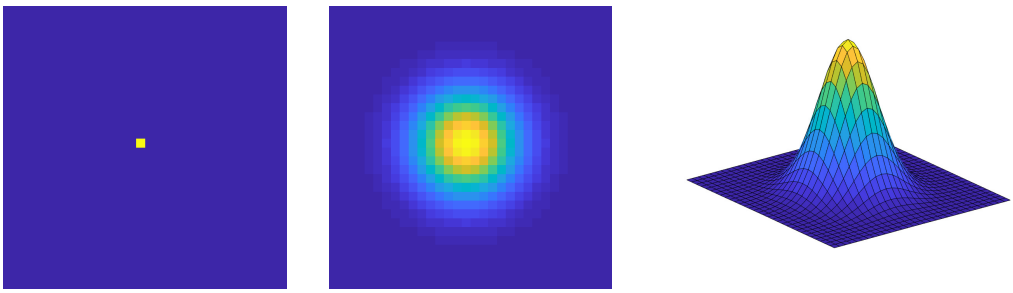
(1.4)

$$[P_k]_{ij} = \gamma \exp\left(-\frac{1}{2}\begin{pmatrix} i - \text{floor}(k/\sqrt{n}) + 1 \\ j - \text{mod}(k,\sqrt{n}) \end{pmatrix}^T \begin{pmatrix} s_1^2 & \rho^2 \\ \rho^2 & s_2^2 \end{pmatrix}^{-1} \begin{pmatrix} i - \text{floor}(k/\sqrt{n}) + 1 \\ j - \text{mod}(k,\sqrt{n}) \end{pmatrix}\right),$$

where $\text{floor}(c)$ is the greatest integer that is less than or equal to c and $\text{mod}(c,d)$ is the remainder after division of c by d. In (1.4) s_1, s_2 and ρ are parameters that define the

orientation and width of the Gaussian PSF, and γ is a normalization constant so that the sum of all the elements of P_k is 1.

Note that the behaviour of the pixels neighbouring the boundaries of a blurred image cannot be fully determined using the model above, as the image is not measured outside of its boundaries. One way of remedying this is to impose suitable boundary conditions, i.e., assumptions on the unmeasured area surrounding the image. The three most classical instances of this are zero, periodic and reflecting boundary conditions; see [HNO06, Section 3.5] and Figure 1.4. Those are also the three boundary condition options that are available for deblurring problems in the IR Tools toolbox [GHN18] (at the date of writing). Other more sophisticated and modern boundary conditions have been developed over the years in order to avoid artifacts in the restored images, for example the anti-reflective boundary conditions [DEMSC06]. As briefly stated in the introduction of this section, different boundary conditions in the forward and inverse problems for synthetic examples can be used to avoid committing an inverse crime.



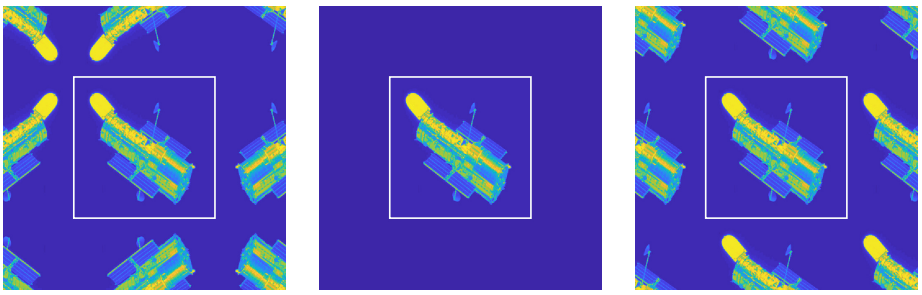
(a) (b) (c)

Figure 1.3: Small Gaussian blur example with $\sqrt{n} = 32$. (a) corresponds to the image of a point source in the central pixel of the image. (b) and (c) display two different representations of the blurred image of the point source in (a); namely a discrete PSF matrix.

We typically assume that the reconstructed image has the same number of pixels as the measurement, so that, for spatially-invariant blurs, the matrix $A \in \mathbb{R}^{n\times n}$ in (1.2) is a square matrix defined by the PSF (describing the transformation of each of the pixels through the blurring), and the boundary conditions. Particular choices of PSFs and boundary conditions lead to different matrix structures for A, which can be exploited directly or used to perform efficient matrix-vector products with A. Moreover, these choices will determine the normality of the matrix A. For example, blurring matrices A coming from a very skewed PSF (e.g., example in Figure 1.1) with reflexive boundary conditions will be highly non-normal, while doubly-symmetric PSFs (e.g., example in Figure 1.2) will lead to a normal A using the same boundary conditions; see [HNO06, Chapter 4] for more details.

1.2.2 Computed Tomography

Computed tomography (CT) is an X-ray imaging technique that aims to reconstruct an object from a set of projections, i.e., measurements obtained by the integration along rays that penetrate a given domain in straight lines. In particular, computed tomography problems can be formulated in the general form (1.1) where the domain Ω is rectangular,



(a) (b) (c)

Figure 1.4: Example of the three most common boundary conditions for image deblurring problems applied to the *satellite* image [GHN18]: (a) reflective boundary conditions; (b) zero boundary conditions; (c) periodic boundary conditions.

the unknown function $f(t)$ represents the image of some material parameter such as the density, and the kernel $k(s,t)$ models the measurement process. For different position angles between the source-detector set up and the object, a set of measurements is taken. There are two main possible geometries for the measuring process at each angle. On the one hand, if the rays are parallel (and usually equidistantly spaced) and the detector is flat, we have parallel-beam tomography. This is the case in synchrotron X-ray measurements, and, mathematically, this process corresponds to the Radon transform. On the other hand, the measurements can be taken by a curved detector from a fan of X-rays coming from a single point source (usually, with an equiangular span between the rays). This is the case in many large medical X-ray scanners. A schematic example of a CT problem with parallel-beam geometry can be observed in Figure 1.5, while a schematic example of the corresponding measuring process can be observed in Figure 1.6. In particular, each column of the sinogram in Figure 1.5 (right image), is constructed by a full set of projections along parallel rays given a fixed position of the phantom (left image), in this case 362 horizontal rays; and each row corresponds to the phantom being rotated by a small angle, in this case by 180 different angles spanning from 0° to 179°.

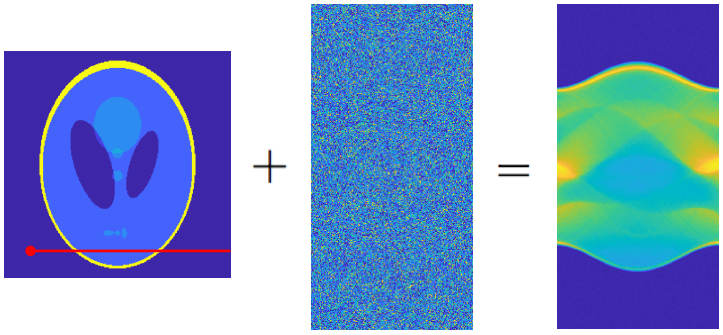


Figure 1.5: Sketch of a tomography problem. On the left, we can observe the original *Shepp Logan* phantom image, available in the MATLAB Image Processing Toolbox [Mat]. The line crossing this image represents one ray. The measurements in the right-hand side correspond to the projections over rays like the one depicted in the leftmost image, after white noise has been added. Another sketch of the measurement process can be observed in Figure 1.6. Note that the sizes of the image and the measurement are different.

To understand how A can be constructed, we consider a basic 2D model problem on the square domain $\Omega = [0,1]^2$. Starting from the integral equation (1.1), a discretization of problem (1.1) can be achieved by considering Ω to be the space of $\sqrt{n} \times \sqrt{n}$ pixel arrays, and $f(t)$ to be piece-wise constant over every pixel. Each entry of the measurements vector

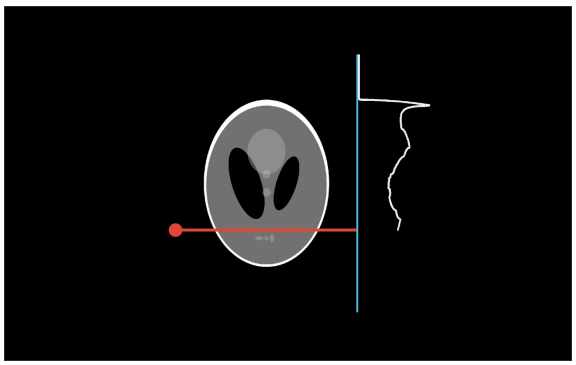


Figure 1.6: Schematic representation of the measuring process for an X-ray acquisition of the Shepp-Logan phantom. Here, the horizontal (red) ray is generated from a source (red circle) and propagates in the right direction. After penetrating the body, where it gets partially absorbed, it is measured when it hits the receiver (vertical blue line). The profile density on the right of the vertical line corresponds to the total amount of absorbed light in each of the horizontal projections corresponding to rays parallel to the one depicted. The density profile depicted on the right corresponds to one column of the sinogram displayed in Figure 1.5, on the right. Image taken from [Sil].

b would then correspond to the measurement taken by a different ray, so that:

$$b_j = \sum_{\substack{k=1,\dots,\sqrt{n} \\ l=1,\dots,\sqrt{n}}} f_{kl} \cdot \Delta L_{kl,j}, \tag{1.5}$$

where $\Delta L_{kl,j}$ is the length of the section of the ray j in the pixel with coordinates (k,l), and f_{kl} is the value of the function $f(t)$ in that pixel. Note that $\Delta L_{kl,j}$ is zero if the ray j does not cross pixel (k,l), so in principle the sum in (1.5) can be very sparse. We can then build the matrix $A \in \mathbb{R}^{m\times n}$ in (1.2) as

$$[A]_{ji} = \begin{cases} \Delta L_{kl,j} & \text{if ray } j \text{ crosses the pixel with coordinates} \\ & (k,l) = (\text{mod}(i,\sqrt{n}),\ \text{floor}(i/\sqrt{n}) + 1) \\ 0 & \text{otherwise} \end{cases},$$

where $\text{floor}(c)$ is the greatest integer that is less than or equal to c and $\text{mod}(c,d)$ is the remainder after division of c by d. Finally, the vector $x \in \mathbb{R}^n$ in (1.2) corresponds to the vectorized image and it is defined component-wise as $x_i = f_{kl}$ for $i = (l-1)\sqrt{n} + k$.

For further details, a good review of the mathematics behind tomography can be found in [Han10], and some really didactic and interactive materials can be found in [Sil].

1.3 Why do we need regularization?

Linear inverse problems of the form (1.2) are often ill-posed in the sense of Hadamard [Had52], i.e., they fulfill at least one of the following undesired properties:

- the problem does not have a solution,
- the problem does not have a unique solution,
- the solutions is very sensitive to small perturbations on the data.

Sometimes, the difficulties associated to problems not having a unique solution, or not having a solution at all, can be alleviated by re-writing the problem as an ℓ_2-norm minimization

$$\min_x \|Ax - b\|_2^2, \tag{1.6}$$

also known as a least squares problem, or by adding additional constraints to the solution. However, it is usually the case that problems of the form (1.2) satisfy the third property, so that small perturbations in the measurements can be arbitrarily amplified in the recovered solution. Therefore, the solutions obtained by straightforwardly solving the linear system (1.2) or the least squares problem (1.6) are very frequently meaningless, and one needs to use regularization, i.e., replace the original problem with a closely related one whose solution is less sensitive to perturbations in the data.

To analyze linear discrete ill-posed problems of the form (1.2) one can resort, at least theoretically, to the singular value decomposition (SVD) of A [GL96, Chapter 2], i.e.,

$$A = \hat{U}\hat{\Sigma}\hat{V}^T, \tag{1.7}$$

where the columns of the square orthogonal matrices $\hat{U} = [\hat{u}_1, .., \hat{u}_m] \in \mathbb{R}^{m\times m}$ and $\hat{V} = [\hat{v}_1, .., \hat{v}_n] \in \mathbb{R}^{n\times n}$ are the left and right singular vectors of A, and the diagonal elements of the rectangular matrix $\hat{\Sigma} = \text{diag}(\hat{\sigma}_1, ..., \hat{\sigma}_{\min\{m,n\}}) \in \mathbb{R}^{m\times n}$, $\hat{\sigma}_i \geq \hat{\sigma}_{i+1} \geq 0$, are its singular values. In the context of ill-posed problems, the singular values of A decay and cluster at 0. If the decay of the singular values is exponential, problem (1.2) is said to be severely ill-posed, whereas if the singular values have power decay, problem (1.2) is said to be moderately or mildly ill-posed; see [Hof86, Chapter 2].

Given the exact measurement $b_{true} = b - e$, defined below equation (1.2), and assuming $Ax_{true} = b_{true}$, the following equality holds:

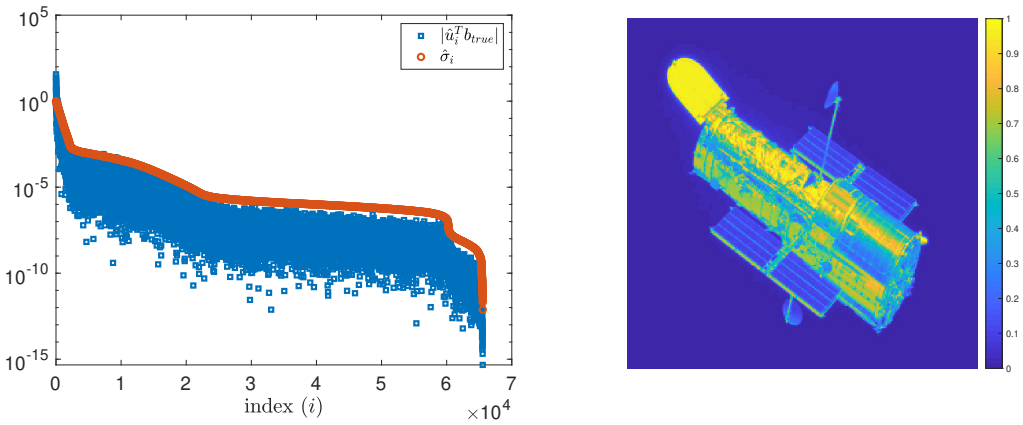
$$x_{true} = A^\dagger b_{true} = \sum_{i=1}^{\min\{m,n\}} \frac{\hat{u}_i^T b_{true}}{\hat{\sigma}_i}\hat{v}_i, \tag{1.8}$$

where $A^\dagger$ is the Moore-Penrose pseudoinverse of A [GL96, Chapter 5]. Since x_{true} is well defined, and the singular values of A decay and cluster at 0, the values of $|\hat{u}_i^T b_{true}|$ need to decay on average at least as fast as $\hat{\sigma}_i$. This is known as the discrete Picard condition [Han90], and can be observed in Figure 1.7 for the problem introduced in Figure 1.2.

Consider now problem (1.2) where the measurements b are corrupted with some noise e. Using the SVD of A, the unregularized solution to (1.2) can be computed as

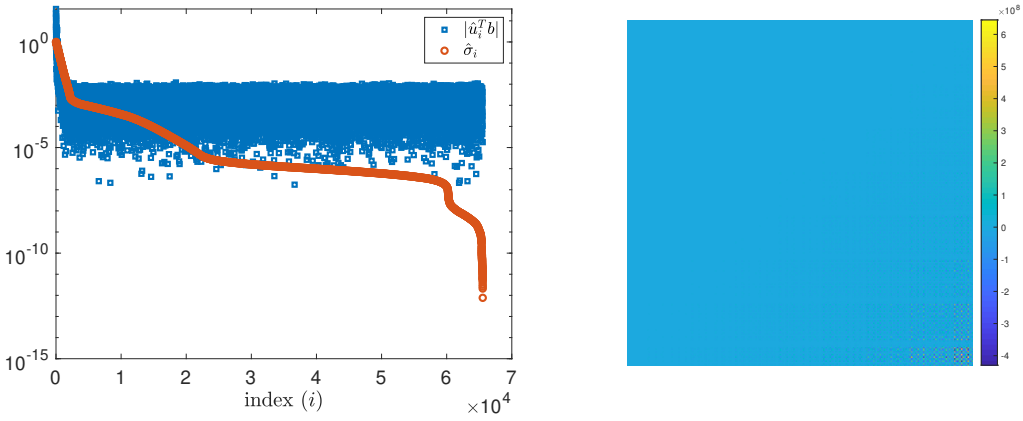
$$x = A^\dagger b = \sum_{i=1}^{\min\{m,n\}} \frac{\hat{u}_i^T b}{\hat{\sigma}_i}\hat{v}_i = \sum_{i=1}^{\min\{m,n\}} \frac{\hat{u}_i^T b_{true}}{\hat{\sigma}_i}\hat{v}_i + \sum_{i=1}^{\min\{m,n\}} \frac{\hat{u}_i^T e}{\hat{\sigma}_i}\hat{v}_i\,, \tag{1.9}$$

where the second term on the right-hand side describes how the error in the measurements propagates in the solution x. Note that this term dominates the solution if $|\hat{u}_i^T e|$ does not decay on average as fast as the singular values of A. For example, this is the case when e is Gaussian white noise, as $|\hat{u}_i^T e|$ is roughly constant; for more details see [Han10, Chapter 4] and the references therein. This behavior can be observed in Figure 1.8 where artificial noise is added in the example shown in Figure 1.7 to simulate the effect described in (1.9). A very powerful tool to obtain a meaningful approximation of the solution of the unavailable noise-free linear system is regularization.



(a) (b)

Figure 1.7: Illustration of the discrete Picard condition for the noise-less version of the deblurring problem presented in Figure 1.2 (i.e., in equation (1.2) we consider x_{exact} and b_{exact}). In this example, $A \in \mathbb{R}^{65536\times 65536}$, but has an exploitable structure (namely, a separable PSF) that allows for an easy computation of the SVD, see [HNO06] for more details. (a) displays the values of $|\hat{u}_i^T b_{true}|$ and $\hat{\sigma}_i$ that appear in equation (1.8): it can be observed how the values of $|\hat{u}_i^T b_{true}|$ for $i = 1, ..., n$ decay on average faster than $\hat{\sigma}_i$. (b) displays x_{true}, which fulfills equation (1.8).



(a) (b)

Figure 1.8: The noisy version of the problem depicted in Figure 1.2 , and whose noise-less version is presented in Figure 1.7, fails the discrete Picard condition. Note that now Gaussian white noise with $nl = 0.01$ perturbs the right-hand side. (a) displays the values of $|\hat{u}_i^T b|$ and $\hat{\sigma}_i$ that appear in equation (1.9); it can be observed how the values of $|\hat{u}_i^T b|$ stabilize so they do not decay on average as fast as $\hat{\sigma}_i$. (b) displays x as computed in (1.9), which is now completely dominated by the noise.

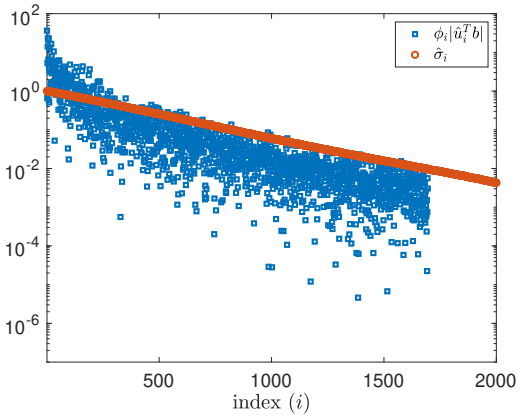
1.4 Direct methods and variational regularization

Direct regularization methods, i.e., those where the regularized solution can be computed directly using an analytic expression, are conceptually the easiest approach to solving inverse problems; see [Han10] for more details. The two classical examples of direct methods are truncated SVD (TSVD) and Tikhonov regularization. For the explanations in this section, recall the definition of the SVD of A (1.7), where $\hat{u}_i$ (resp. $\hat{v}_i$) denote the left (resp. right) singular vectors of A, and $\hat{\sigma}_i$ are the singular values of A.

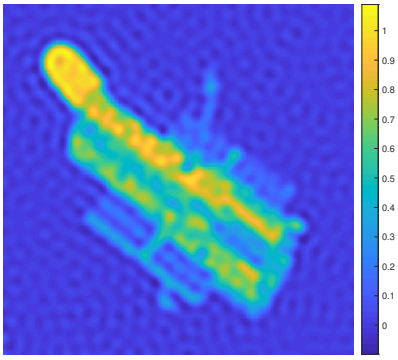
The truncated SVD (TSVD) seeks a solution of the form:

$$x_k^{tsvd} = \sum_{i=1}^{k \ll n} \frac{\hat{u}_i^T b}{\hat{\sigma}_i} \hat{v}_i = \sum_{i=1}^{n} \frac{\phi_i(k) \hat{u}_i^T b}{\hat{\sigma}_i} \hat{v}_i \,, \quad \phi_i(k) = \begin{cases} 1 & \text{for } i \leq k, \\ 0 & \text{otherwise} \end{cases}. \tag{1.10}$$

Looking at the second equality in equation (1.10), we can observe that the TSVD solution can be written in a similar form to equation (1.9), where a factor $\phi_i(k)$ that depends on the truncation parameter k is multiplied in each of the terms of equation (1.9). Since the basis given by the SVD can be considered a spectral basis [Han10, Section 2.1], the TSVD method in expression (1.10) can be considered a spectral filtering method, and so the scalars $\phi_i(k)$ are usually called filter factors. As an example, Figure 1.9 (b) displays the TSVD reconstruction for the test problem in Figure 1.2, while Figure 1.9 (a) shows the corresponding values of $\phi_i(k)|\hat{u}_i^T b|$ and $\hat{\sigma}_i$, which can be compared to the ones displayed in Figure 1.8 (a). The truncation parameter k acts as a discrete regularization parameter for TSVD, and therefore the choice of k determines the success of TSVD as a regularization method. When solving an ill-posed problem, TSVD with k too small will produce an over-regularized solution (with a big residual norm); while TSVD with k too big will produce an under-regularized solution (with a big solution norm as the last factors in the sum (1.10) for $i \leq k$ might satisfy the following undesired property: $|\hat{u}_i^T b| >> \hat{\sigma}_i$).



(a)



(b)

Figure 1.9: Illustration corresponding to the TSVD method, with truncation parameter $k = 1695$, applied to the deblurring problem presented in Figure 1.2 with $nl = 0.01$. (a) displays the values of $\phi_i(k)|\hat{u}_i^T b|$ and $\hat{\sigma}_i$ that appear in equation (1.10). It can be observed how the values of $|\hat{u}_i^T b|$ decay on average as fast as $\hat{\sigma}_i$ for $i \leq k = 1695$. (b) displays the corresponding TSVD solution (1.10).

Tikhonov regularization is defined as:

$$x_{\lambda,L}^{Tik} = \arg\min_{x \in \mathbb{R}^n} \{ \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2 \} = (A^T A + \lambda L^T L)^{-1} A^T b \,, \quad \lambda \geq 0, \tag{1.11}$$

where $L \in \mathbb{R}^{q \times n}$ is the regularization matrix and λ is the regularization parameter that balances the effect of the fit-to-data term $\|Ax - b\|_2^2$ and the regularization term $\|Lx\|_2^2$. The choice of L encourages desired known properties on the recovered solution by penalizing the norm of unwanted features arising in the product Lx. The most natural choices for L are either the identity matrix, which penalizes a big solution norm (and problem (1.11) is said to be in standard form), or a discrete approximation to a derivative operator. Since high frequency components are magnified with differentiation, choosing the regularization matrix to be a discrete approximation to a derivative operator penalizes high frequency components, thus promoting smoother solutions [HNO06, Chapter 7]. Note that, when

$L \in \mathbb{R}^{q\times n}$ is a discrete approximation to a derivative operator of order o in d dimensions, $q = d(n-o)$, so that the properties of problem (1.11), and suitable implementation strategies to solve (1.11), heavily depend on the number of dimensions of the problem. We will assume that L is chosen so that $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$, where $\mathcal{N}(B)$ denotes the null space of a matrix B, so that problem (1.11) has a unique solution and the closed form expression after the second equality in (1.11) is well defined. Analogously to the choice of a good truncation parameter k in the TSVD, the choice of a good regularization parameter λ is crucial for the success of Tikhonov regularization (1.11). When solving an ill-posed problem, Tikhonov regularization with λ too big produces over-regularized solutions (with a big discrepancy in the fit-to-data term), while Tikhonov regularization with λ too small produces under-regularized solutions (with a big solution norm). Chapter 2 offers a review of existing parameter choice rules for large-scale problems and presents a new framework to obtain a good regularization parameter λ in the context of Krylov projection methods. More specifically, this is presented in the paper [GL20] included in this thesis.

Tikhonov regularization is a spectral filtering method. Analogously to TSVD, and for $L = I$, equation (1.11) can be re-written in a similar form to (1.9):

$$x_{\lambda,I}^{Tik} = \sum_{i=1}^{n} \frac{\phi_i(\lambda)\hat{u}_i^T b}{\hat{\sigma}_i}\hat{v}_i, \qquad \phi_i(\lambda) = \frac{\hat{\sigma}_i^2}{\hat{\sigma}_i^2 + \lambda} \leq 1. \tag{1.12}$$

An example of this method can be observed in Figure 1.10. Note that equation (1.12) can be generalized to $L \neq I$ using the generalized SVD (GSVD) of the matrix pair $\{A,\ L\}$ [GVL96]. Tikhonov problems in general form, i.e., when $L \neq I$, can also be approached by performing a standard form transformation to (1.11); see [Eld82]. Indeed, using the A-weighted pseudoinverse of L,

$$L_A^\dagger = [I - (A(I - L^\dagger L))^\dagger A]L^\dagger \in \mathbb{R}^{n\times q},$$

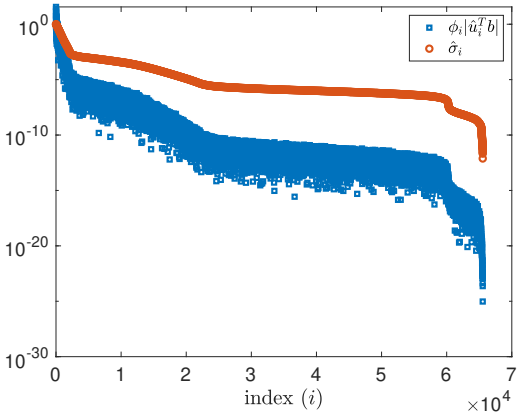
the solution $x_{\lambda,L}^{Tik}$ to equation (1.11) can be decomposed as the sum of a vector $x_{0,L} \in \mathcal{N}(L)$ and a vector in the range of $L_A^\dagger$, so that $x_{\lambda,L}^{Tik} = L_A^\dagger \bar{x}_{\lambda,L}^{Tik} + x_{0,L}$. Considering the transformations $\bar{A} = AL_A^\dagger$ and $\bar{b} = b - Ax_{0,L}$, (1.11) can be equivalently re-written as a standard form Tikhonov problem:

$$\bar{x}_{\lambda,L}^{Tik} = \arg\min_{\bar{x}\in\mathbb{R}^n} \{ \left\|\bar{A}\bar{x} - \bar{b}\right\|_2^2 + \lambda \left\|\bar{x}\right\|_2^2 \}, \quad x_{\lambda,L}^{Tik} = L_A^\dagger \bar{x}_{\lambda,L}^{Tik} + x_{0,L},$$

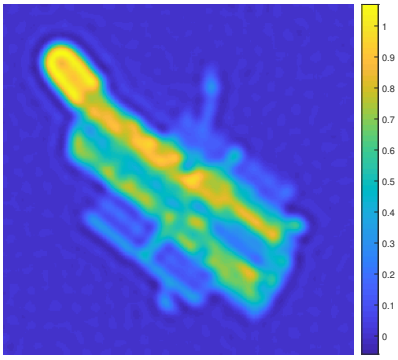
where $x_{0,L} = (A(I - L^\dagger L))^\dagger b$. This approach is extensively used in the papers [GL20] and [GL19], included in Chapter 2 and 3 respectively. Note that, when L is square and non-singular $L_A^\dagger = L^{-1}$ and $x_{0,L} = 0$; also, when $q \geq n$ and $L \in \mathbb{R}^{q\times n}$ is full-rank, then $L_A^\dagger = L^\dagger$ and $x_{0,L} = 0$.

Tikhonov regularization as defined in (1.11) has a limited performance. For example, since using 2-norm regularization terms promotes a smooth solution, Tikhonov regularization will have a poor performance when the exact solution x_{true} has non-smooth components, e.g., x_{true} is an image displaying sharp edges or x_{true} is sparse. For more details see, e.g., [Han10, Section 8.6] or [Vog02, Section 1.2]. Therefore, we consider more advanced regularization schemes that incorporate knowledge about the object that we want to reconstruct or the noise in the measurements. We focus on variational regularization methods for (1.2) [SGG+08, Chapter 3], which can be written in the following general form

$$\min_{x\in\mathbb{R}^n} \{\rho(Ax, b) + \lambda R(x)\}, \quad \lambda \geq 0, \tag{1.13}$$



(a)



(b)

Figure 1.10: Illustration corresponding to the Tikhonov method in standard form, with $\lambda = 8.4 \cdot 10^{-4}$, applied to the deblurring problem presented in Figure 1.2 with $nl = 0.01$. (a) displays the values of $\phi_i(\lambda)|\hat{u}_i^T b|$ and $\hat{\sigma}_i$ that appear in equation (1.12). It can be observed how the values of $\phi_i(\lambda)|\hat{u}_i^T b|$ decay on average faster than $\hat{\sigma}_i$. (b) displays the corresponding Tikhonov solution (1.12).

where, as in (1.11), the regularization parameter λ balances the effect of the fit-to-data term $\rho(Ax, b)$, measuring the error between Ax and the (often noisy) measurement b, and the regularization term $R(x)$. Note that Tikhonov regularization (1.11) can be written in the form (1.13), as well as TSVD, by defining a suitable function $\rho(Ax, b)$ involving the SVD of A. A very popular choice for the fit-to-data term is

$$\rho(Ax, b) = \|M(Ax - b)\|_r^r,$$

where M is formally a preconditioning matrix that, in the inverse problems setting, can be used to incorporate prior information about the noise; see [Cal07]. For example, for colored Gaussian noise whose entries are uncorrelated but have different variances, i.e., with covariance matrix $C_e = \text{diag}(\sigma_1^2, ..., \sigma_m^2)$, a good choice for the preconditioning matrix is $M = \text{diag}(1/\sigma_1, ..., 1/\sigma_m)$ and $r = 2$, see, e.g., [RHM10]. Moreover, similar preconditioning techniques can be used to consider other noise distributions, for example, this is used in [BN06] for CCD camera noise statistics that describe the noise distribution as a sum of Poisson and Gaussian distributions. In particular, when the noise in the measurements e is Gaussian white noise, a good choice of M is a rescaled version of the identity matrix, and $r = 2$, so we are left with the following problem:

$$\min_{x \in \mathbb{R}^n} \{\|Ax - b\|_2^2 + \lambda R(x)\}, \quad \lambda \geq 0, \tag{1.14}$$

where possible scaling factors have been absorbed by the regularization parameter λ. The general ℓ_r-ℓ_p regularization problem is a particular case of (1.13), which is defined as:

$$\min_{x \in \mathbb{R}^n} \{\|M(Ax - b)\|_r^r + \lambda \|Lx\|_p^p\}. \tag{1.15}$$

In particular, for $M = I$, $r = 2$ and $L = I$,

$$x_\lambda^{\ell_p} = \arg\min_{x \in \mathbb{R}^n} \{\|Ax - b\|_2^2 + \lambda \|x\|_p^p\}, \tag{1.16}$$

and (1.16) is widely used to compute sparse solutions of linear discrete ill-posed problems when $0 < p \leq 1$. Indeed, the ℓ_0 "norm", which amounts to the number of non-zero elements of a vector, is small by definition for sparse vectors. As the minimization problem (1.16)

for $p = 0$ is NP hard [FR11], it is common to approximate it using $0 < p \leq 1$. However, for $0 < p < 1$, the ℓ_2-ℓ_p regularization problem is non-convex, so the minimization problem (1.16) can produce non-meaningful reconstructions due to local minima of the objective function in (1.16). A well-established method to compute sparse solutions of linear discrete ill-posed problems is ℓ_2-ℓ_1 regularization, as it is the convex relaxation of the ℓ_2-ℓ_0 problem. Note that some care has to be taken in the convex minimization of the functional in (1.16) as it is not differentiable at the origin for $p = 1$. For example, in signal processing, this minimization scheme is usually exploited in the area of compressed sensing [FR13]. Finally, note that the ℓ_2-ℓ_p regularization problem (1.16) can be formulated as a non-linear weighted least squares problem of the form:

$$x_\lambda^{\ell_p} = \arg\min_{x\in\mathbb{R}^n} \{\|Ax - b\|_2^2 + \lambda\|x\|_p^p\} = \arg\min_{x\in\mathbb{R}^n} \{\|Ax - b\|_2^2 + \lambda\|W^{(p)}(x)x\|_2^2\}, \tag{1.17}$$

where

$$W^{(p)}(x) = \operatorname{diag}\left((|[x]_i|^{\frac{p-2}{2}})_{i=1,\dots,n}\right). \tag{1.18}$$

A well-established framework to solve problem (1.16), usually referred to as iteratively reweighted least squares (IRLS) [DDFG10] or iteratively reweighted norm method (IRN) [WR08a], is the local approximation of (1.17) by a sequence of quadratic Tikhonov problems (1.11) of the form

$$x_k = \arg\min_{x\in\mathbb{R}^n} \{\|Ax - b\|_2^2 + \lambda\|W_k x\|_2^2\}, \quad k = 1, 2, ..., \tag{1.19}$$

where the sequence of weights W_k is updated using the available approximate solution x_{k-1} from the previous problem in the sequence. Note that, when $0 < p < 2$, some care has to be taken to avoid division by zero when evaluating the weights (1.18) at an approximate solution with $[x]_i = 0$ for at least one $1 \leq i \leq n$: this is likely to happen as ℓ_2-ℓ_p regularization promotes sparse approximate solutions of (1.2). Division by zero can be avoided by either adding a suitable threshold on the definition of $W_k = (W^{(p)}(x_{k-1})^{-1} + \tau)^{-1}$, or by replacing (1.17) by a smooth approximation of the ℓ_2-ℓ_p regularization problem (whose corresponding functional is differentiable at the origin) with the following weights

$$\widetilde{W}^{(p,\tau)}(x) = \operatorname{diag}\left(((|[x]_i^2 + \tau^2)^{\frac{p-2}{4}})_{i=1,\dots,n}\right), \tag{1.20}$$

so that $W_k = \widetilde{W}^{(p,\tau)}(x_{k-1})$. If we consider the smooth approximation of ℓ_2-ℓ_p regularization with weights defined in (1.20), solving the sequence (1.19) with $W_k = \widetilde{W}^{(p,\tau)}(x_{k-1})$ is a particular instance of a majorization-minimization (MM) scheme [HLM$^+$17], where (1.19) corresponds to the non-constant terms of a quadratic majorization of the smoothed version of the objective function in (1.17). The paper [GNL21], included in Chapter 3, exploits this framework for seeking sparse solutions of linear inverse problems. Note that generalizations of this method can be applied also when one chooses $M = I$, $r = 2$ and $L \neq I$ in (1.15) to enforce sparsity of the solution in a basis given by the range of L; see [GNL21] and references therein. Paper [GL19], included in Chapter 4, uses this framework to tackle a more sophisticated regularization method called total variation (TV).

Total variation (TV) regularization is another popular regularization method defined as follows,

$$x_\lambda^{TV} = \arg\min_{x\in\mathbb{R}^n} \{\|Ax - b\|_2^2 + \lambda \mathrm{TV}(x)\}, \tag{1.21}$$

where $\mathrm{TV}(x)$ is the discrete isotropic total variation of x and it is defined as the ℓ_1-norm of the discrete gradient of x. Similarly to ℓ_2-ℓ_p regularization, described in (1.16), the ℓ_1 norm

is a sparsity-enforcing norm when used in the regularization term; in this case, it enforces sparsity in the gradient of x. Note that this can be understood as a special case of sparsity under transform; however, special care needs to be taken as the derivative transformation is not invertible. Sparsity in the gradient promotes piece-wise constant reconstructions so, in the context of imaging, this regularization term is very effective to preserve edges. Note that, as we are interested in sparsity-promoting norms, problem (1.21) can also be generalized by TV_p regularization, where the discrete gradient is evaluated in the ℓ_p norm, for $0 < p \leq 1$. As explained for ℓ_2-ℓ_p regularization, choosing $0 < p < 1$ leads to a better approximation to the 0 'quasi-norm' for the regularization term, but it is a non-convex optimization problem.

In the one-dimensional case, $\mathrm{TV}(x) = \|D_{1\mathrm{d}}x\|_1$, where

$$D_{1\mathrm{d}} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times n}. \tag{1.22}$$

In the two-dimensional case, $\mathrm{TV}(x) = \| \left((D^{\mathrm{h}}x)^2 + (D^{\mathrm{v}}x)^2 \right)^{1/2} \|_1$, where $x \in \mathbb{R}^n$ is obtained by stacking the columns of a 2D array $X \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$ as defined in (1.3). The discrete first derivatives in the horizontal and vertical directions are given by

$$D^{\mathrm{h}} = (D_{1\mathrm{d}} \otimes I) \in \mathbb{R}^{\sqrt{n}(\sqrt{n}-1)\times n}, \qquad D^{\mathrm{v}} = (I \otimes D_{1\mathrm{d}}) \in \mathbb{R}^{\sqrt{n}(\sqrt{n}-1)\times n},$$

respectively. Here $D_{1\mathrm{d}} \in \mathbb{R}^{(\sqrt{n}-1)\times\sqrt{n}}$ is the 1D first derivative matrix (1.22) of appropriate size, and I is the identity matrix of size $\sqrt{n}$, so that 2D discrete operators are defined in terms of the corresponding 1D ones (see [GL19] for more details).

Since the $\mathrm{TV}(x)$ regularization functional is non-differentiable, total variation regularization as defined in (1.21) requires solving a convex but non-smooth minimization problem. Analogously to ℓ_2-ℓ_p regularization, TV regularization can be re-written as a weighted non-linear least squares problem:

$$\min_{x\in\mathbb{R}^n} \{\|Ax - b\|_2^2 + \lambda \mathrm{TV}(x)\} = \min_{x\in\mathbb{R}^n} \{\|Ax - b\|_2^2 + \lambda \|W(Dx)Dx\|_2^2\}. \tag{1.23}$$

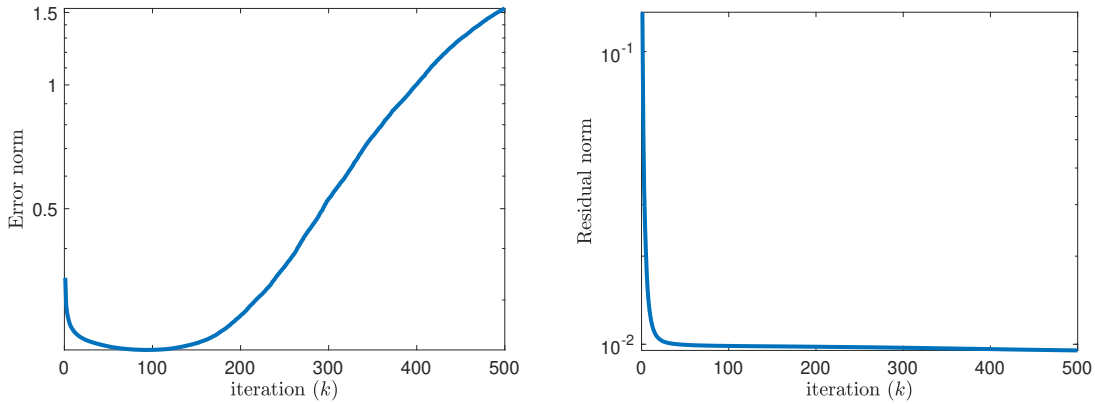
This can be solved using an IRLS or and IRN scheme [WR08b], so (1.23) is approximated by a sequence of quadratic problems; each of them replacing the solution-dependent weights $W(Dx)$ in the $\mathrm{TV}(x)$ functional by the weights evaluated at the approximated solution of the previous problem in the sequence. As for ℓ_2-ℓ_p regularization, suitable modifications of $W(Dx_{k-1})$ must be considered to avoid division by zero. The formal definition of the weights $W(Dx)$ is different if the problem is in 1D or 2D, and D is a scaled finite differences matrix that discretizes a first order derivative operator in either 1D or 2D. More details can be found in Chapter 4 and the included paper [GL19], where we also present a new algorithm called TV-FGMRES to solve total variation regularization problems when $A \in \mathbb{R}^{n\times n}$.

1.5 Iterative regularization and Krylov Methods

Storing and accessing the coefficient matrix A associated to large-scale problems, like those naturally arising from imaging applications, is computationally very demanding. In these cases, unless A has an exploitable structure, direct methods that require expensive computations such as determining the SVD of A are not feasible. Moreover, A is sometimes

coded as a function that efficiently performs matrix-vector products between A (or possibly A^T) and arbitrary vectors v, but it cannot be accessed explicitly: this scenario is usually referred to as a matrix-free problem. In these cases, one has to resort to an iterative method to find a solution for problem (1.2).

Many iterative methods can be used as regularization methods because of their semi-convergent behavior, i.e., the computed solution initially approximates the solution of the noise-free problem $A^\dagger b_{true}$ and then deteriorates due to the noise e present in the measurements, as it starts to approximate $A^\dagger b$. For this reason, a clever stopping criterion for the iterations of iterative solvers for linear systems has to be decided, and the total amount of iterations k^* can be considered the regularization parameter for the method. An illustration of this phenomenon can be observed in Figure 1.11 for LSQR, an iterative Krylov subspace method that will be described in the following paragraphs.



(a) (b)

Figure 1.11: LSQR method applied to the deblurring problem presented in Figure 1.2 with $nl = 0.01$. The semi-convergence behaviour can be observed in these plots: in (a) the value of the relative error norm $\|x_k - x_{exact}\|_2/\|x_{exact}\|_2$ decreases in the first iterations and then starts increasing again, while in (b) the relative residual norm $\|Ax_k - b\|_2/\|b\|_2$ decreases monotonically due to the optimality properties of LSQR.

One of the most basic iterative linear system solvers is gradient descent method. At iteration k, given an initial guess x_0 and a step length α_k that can be iteration-dependent, the gradient descent method computes an approximation of the solution of (1.2) of the form

$$x_{k+1} = x_k + \alpha_k A^T(b - Ax_k). \tag{1.24}$$

There are a variety of solvers that can be written in the form (1.24), and that differ on the choice of α_k, e.g., the Landweber algorithm determines α_k as a relaxation parameter [EHN96, Chapter 6.1]. However, even for good choices of α_k, the gradient descent method typically displays very slow convergence. A very powerful alternative is the use of Krylov subspace methods, which are projection methods onto Krylov subspaces. Given a matrix $C \in \mathbb{R}^{n\times n}$ and a vector $d \in \mathbb{R}^n$, a Krylov subspace of dimension k is defined as

$$\mathcal{K}_k(C, d) = \text{span}\{d, Cd, \ldots, C^{k-1}d\}.$$

In the most general case, when applying a Krylov method at iteration k, we seek a solution x_k such that:

$$x_k \in \mathcal{K}_k(C, d), \quad r_k = b - Ax_k \perp \mathcal{K}_k(C', d'). \tag{1.25}$$

In particular, we are interested in the case where $\mathcal{K}_k(C', d') = A\mathcal{K}_k(C, d)$, so that x_k fulfills the conditions in (1.25) if and only if

$$x_k = \arg \min_{x \in \mathcal{K}_k(C,d)} \|b - Ax\|_2. \tag{1.26}$$

For this reason, these methods are usually called minimal residual Krylov methods [Saa03, Section 5]. We will focus on the two most common examples of such methods. On the one hand, for $A \in \mathbb{R}^{n\times n}$, choosing $C = A$ and $d = b$ leads to GMRES. On the other hand, for a general A, choosing $C = A^TA$ and $d = A^Tb$ leads to LSQR, which is mathematically equivalent to CGLS. From now on only minimal residual Krylov methods will be considered.

Krylov methods are iterative regularization methods based on partial factorizations of A (and sometimes A^T) that are updated at each iteration. Golub-Kahan bidiagonalization and the Arnoldi algorithm provide suitable partial factorizations to be used in LSQR and GMRES (respectively). Both are based on adaptations of a Gram-Schmidt-like orthogonalization process and will be detailed in the following.

The Golub-Kahan bidiagonalization algorithm and LSQR

Golub-Kahan bidiagonalization [GK65], also known as the Lanczos bidiagonalization algorithm, computes orthogonal bases for the Krylov subspaces $\mathcal{K}_k(A^TA, A^Tb)$ and $\mathcal{K}_k(AA^T, b)$, which are spanned by the column vectors of the matrices $V_k \in \mathbb{R}^{n\times k}$ and $U_k \in \mathbb{R}^{m\times k}$, respectively. The partial factorizations of A, and A^T, computed by the Golub-Kahan bidiagonalization algorithm are

$$\begin{aligned} AV_k &= U_{k+1}\bar{B}_k, \\ A^TU_{k+1} &= V_k\bar{B}_k^T + \rho_{k+1}\zeta_{k+1}e_{k+1}^T, \end{aligned} \tag{1.27}$$

where

$$\bar{B}_k = \begin{bmatrix} \rho_1 & & & \\ \zeta_2 & \rho_2 & & \\ & \ddots & \ddots & \\ & & \zeta_k & \rho_k \\ & & & \zeta_{k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k},$$

and e_{k+1} denotes the $(k+1)$th canonical basis vector of $\mathbb{R}^{k+1}$. Note that the first column of U_k is chosen to be $b/\|b\|$. Algorithm 1 reports a summary of a basic implementation of the Golub-Kahan bidiagonalization procedure without re-orthogonalization. In practice one should re-orthogonalize (at least) the columns of U_k or V_k [Bar13] to avoid numerical instabilities and to ensure the spectral properties of the matrix $\bar{B}_k$ associated to the projected problem approximate the ones of matrix A [RVEA17]. Note that the computational cost of this method is dominated by a matrix-vector product with A and a matrix-vector product with A^T at each iteration, and that explicitly storing A is not needed.

The partial factorization (1.27) given by the Golub-Kahan bidiagonalization algorithm is used in LSQR to update the solution at each iteration by solving the minimization problem (1.26). More specifically, the approximated solution to problem (1.6) belonging to $\mathcal{K}_k(A^TA, A^Tb)$ can be written as $x_k = V_ky_k$, where y_k is determined by projecting (1.6):

$$y_k = \arg \min_{y\in\mathbb{R}^k} \|b - AV_ky\|_2 = \arg \min_{y\in\mathbb{R}^k} \|b - U_{k+1}\bar{B}_ky\|_2 = \arg \min_{y\in\mathbb{R}^k} \|\|b\|_2e_1 - \bar{B}_ky\|_2. \tag{1.28}$$

Algorithm 1 Golub-Kahan bidiagonalization algorithm

Input: A, b.
Initialize: $\zeta_1 = \|b\|_2$, $u_1 = b/\zeta_1$.
Initialize: $v = A^T u_1$, $\rho_1 = \|v\|_2$, $v_1 = v/\rho_1$.
For $j = 2, \ldots, k+1$

1. Compute $u = Av_{j-1} - \rho_{j-1}u_{j-1}$.
2. Set $\zeta_j = \|u\|_2$.
3. Take $u_j = u/\nu_j$.
4. Compute $v = A^T u_j - \nu_j v_{j-1}$.
5. Set $\rho_j = \|v\|_2$.
6. Take $v_j = v/\rho_j$.

Note that the second equality in (1.28) is obtained using the partial factorization of A in (1.27), and the third equality is obtained using the fact that U_{k+1} has orthogonal columns and that the fist column of U_{k+1} is $u_1 = b/\|b\|_2$, so that $b = U_{k+1}\|b\|_2 e_1$. The computations involved at the kth LSQR iteration are sketched in Algorithm 3.

Note that so far we have assumed that no initial guess is known for the solution, i.e., $x_0 = 0$. However, LSQR can be easily generalized to the case where an initial guess $x_0 \neq 0$ is known.

The Arnoldi algorithm and GMRES

For any square matrix $A \in \mathbb{R}^{n\times n}$, the Arnoldi algorithm constructs an orthonormal basis for the Krylov subspace $\mathcal{K}_k(A, b)$. Let $V_k \in \mathbb{R}^{n\times k}$ be the matrix whose columns span $\mathcal{K}_k(A, b)$. Then the partial decomposition of A given by the Arnoldi algorithm is

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T, \tag{1.29}$$

where e_k is the kth canonical basis vector of $\mathbb{R}^k$ and $H_k \in \mathbb{R}^{k\times k}$ is an upper Hessenberg matrix that represents the orthogonal projection of A onto the Krylov subspace $\mathcal{K}_k(A, b)$. Note that the fist vector of V_k corresponds to a normalized version of the vector of measurements b. Equivalently, relation (1.29) can be written as

$$AV_k = V_{k+1}\bar{H}_k, \quad \text{for} \quad \bar{H}_k = \begin{bmatrix} H_k \\ h_{k+1,k} e_k^T \end{bmatrix} \in \mathbb{R}^{(k+1)\times k}. \tag{1.30}$$

A basic implementation of the Arnoldi algorithm (without re-orthogonalization) is described in Algorithm 2, where it can be observed that its computational cost is dominated by the cost of a matrix-vector product with A. Analogously to Golub-Kahan bidiagonalization, A does not need to be stored explicitly. Note that, in practice, one might want the re-orthogonalize the columns of U_k.

The partial decomposition of A (1.29) given by the Arnoldi algorithm is the main building block of GMRES. At each iteration, GMRES computes an approximated solution to problem (1.6) belonging to $\mathcal{K}_k(A, b)$, $x_k = V_k y_k$, where

$$y_k = \arg\min_{y\in\mathbb{R}^k} \|b - AV_k y\|_2 = \arg\min_{y\in\mathbb{R}^k} \|b - V_{k+1}\bar{H}_k y\|_2 = \arg\min_{y\in\mathbb{R}^k} \|\|b\|_2 e_1 - \bar{H}_k y\|_2. \tag{1.31}$$

Algorithm 2 Arnoldi algorithm

Input: A, b.
Initialize: $v_1 = b/\|b\|_2$.
For $j = 1, 2, \ldots, k$

1. For $i = 1, \ldots, j$: compute $[H]_{i,j} = (Av_j)^T v_i$.
2. Compute $v = Av_j - \sum_{i=1}^{j} [H]_{i,j} v_i$.
3. Define $[H]_{j+1,j} = \|v\|_2$.
4. If $[H]_{j+1,j} = 0$ stop; else take $v_{j+1} = v/[H]_{j+1,j}$.

Here, we have used the partial factorization (1.30) (equivalent to (1.29)), the fact that V_{k+1} has orthogonal columns, and $v_1 = V_{k+1}e_1 = b/\|b\|_2$. A general step of GMRES at iteration k is sketched in Algorithm 3.

Algorithm 3 LSQR and GMRES solution update at iteration k

1. Expand the Krylov subspace by adding a column v_k to the matrix V_k and updating the partial factorizations in (1.27) for LSQR or (1.29) for GMRES.
2. Solve the projected LS problem (1.28) for LSQR and (1.31) for GMRES.
3. Compute the approximation $x_k = V_k y_k$.

Analogously to LSQR, GMRES can be easily generalized to the case where an initial guess for the solution $x_0 \neq 0$ is known. For $A \in \mathbb{R}^{n \times n}$ GMRES can be appealing with respect to LSQR as it is cheaper per iteration, but it can display very bad performance when the system matrix A is highly non-normal.

When the Arnoldi algorithm is applied to Hermitian matrices it can be simplified, so we can perform a three-term recursion when constructing the partial factorization of A: this is known as the symmetric Lanczos algorithm [Saa03, Chapter 6]. In particular, for symmetric matrices A, the upper Hessenberg matrix H_k in (1.29) is symmetric tridiagonal. This is the main building block for both MINRES and CGLS. In fact, the symmetric Lanczos algorithm and the Golub-Kahan bidiagonalization (or Lanczos bidiagonalization algorithm) are closely related, and CGLS is mathematically equivalent to LSQR . This notion is explained and unfolded in Chapter 2, as it is extensively used in the included paper [GL20].

1.5.1 Krylov-Tikhonov Methods

Due to the semi-convergence phenomenon, the performance of Krylov methods heavily relies on a good stopping criterion for the iterations. Moreover, it is possible that semi-convergence happens before all the significant basis vectors for the solution have been incorporated in the Krylov subspace, so a later stopping can improve the quality of the reconstructed solution if the semi-convergence effect is mitigated, i.e., if the error norm stabilizes at the point of semi-convergence; see [Han10, Chapter 6] or [HJ17]. For this reason, it is common to jointly use Krylov projection methods and a direct regularization method, so that the advantages of Krylov methods (efficiency, fast convergence and a good regularizing space for the solution) are preserved while the new algorithm is less sensitive to semi-convergence (which might depend on the choice of a good regularization parameter

for the direct regularization method). In particular, we focus on methods that combine Tikhonov regularization and minimal residual Krylov subspace methods. For simplicity, we will consider $L = I$ in (1.11).

Note that there are two possible ways of combining Tikhonov regularization and minimal residual Krylov subspace methods to obtain "the best of both worlds" [Han10, Section 6.4]. As it can be observed in Figure 1.12, one could either project the original Tikohonov problem (1.11) in standard form onto a Krylov subspace of increasing dimension in a "first-regularize-then-project" approach, or regularize the sequence of projected problems (1.26) with a standard Tikhonov regularization term in a "first-project-then-regularize" approach. The methods in the latter class are also generally known as hybrid methods [CNO08, OS81]. An important feature of Krylov-Tikhonov methods is that, for fixed λ, and due to the orthogonality of the columns of V_k, the problems solved at each iteration of the "first-regularize-then-project" approach and the "first-project-then-regularize" approach are equivalent. This can be observed in Figure 1.12. Due to this equivalence, hybrid methods can be written in an analogous framework to Algorithm 3 where, in step 2, the projected problem (1.28) is replaced by

$$y_k = \arg\min_{y\in\mathbb{R}^k} \{\|b - AV_k y\|_2^2 + \lambda\|V_k y\|_2^2\} = \arg\min_{y\in\mathbb{R}^k} \{\|\|b\|_2 e_1 - \bar{B}_k y\|_2^2 + \lambda\|y\|_2^2\}$$

for hybrid LSQR, and (1.31) in replaced by

$$y_k = \arg\min_{y\in\mathbb{R}^k} \{\|b - AV_k y\|_2^2 + \lambda\|V_k y\|_2^2\} = \arg\min_{y\in\mathbb{R}^k} \{\|\|b\|_2 e_1 - \bar{H}_k y\|_2^2 + \lambda\|y\|_2^2\}$$

for hybrid GMRES.

Finding a good regularization parameter λ for Krylov-Tikhonov methods, when it is not known a-priori, has drawn a lot of attention in recent years. In particular, hybrid methods (corresponding to the "first-project-then-regularize" approach) are very popular in this area of research as it has been observed that if a good regularization parameter is properly chosen at each iteration for each projected sub-problem (so that λ_k changes through the iterations), the value of λ_k as k increases seems to stabilize around a value λ that is also good for the full-dimensional original Tikhonov problem (1.11) [CKO15]. Specifically, examples of automatic parameter choice rules can be found in [RR13, RS08, GN14], and in the references therein. In [GL20], presented in Chapter 2, we derive a new class of rules for the simultaneous choice of the stopping iteration and a good regularization parameter λ_k.

1.5.2 Modified Krylov subspaces

Although Krylov subspace methods have inherent regularizing properties [Han98, Chapter 6], Krylov subspaces can be further modified to enhance particular properties of the reconstructed solution. For example, a-priori information about the solution we want to recover can be incorporated into the Krylov subspaces through prior-conditioning [Cal07], a method that is formally equivalent to preconditioning but that is theoretically motivated by the Bayesian interpretation of the linear system (1.2). Differently from traditional preconditioners, used to accelerate the convergence of iterative methods [Saa03], prior-conditioners can be used to improve the quality of the recovered solution, while maintaining a similar (or a relatively slower) convergence rate. This concept is further detailed below.

In the Bayesian framework, a statistical interpretation is given for the linear system (1.2), i.e., x, b and e are modeled as random variables [CS07, Chapter 2]. When the distribution

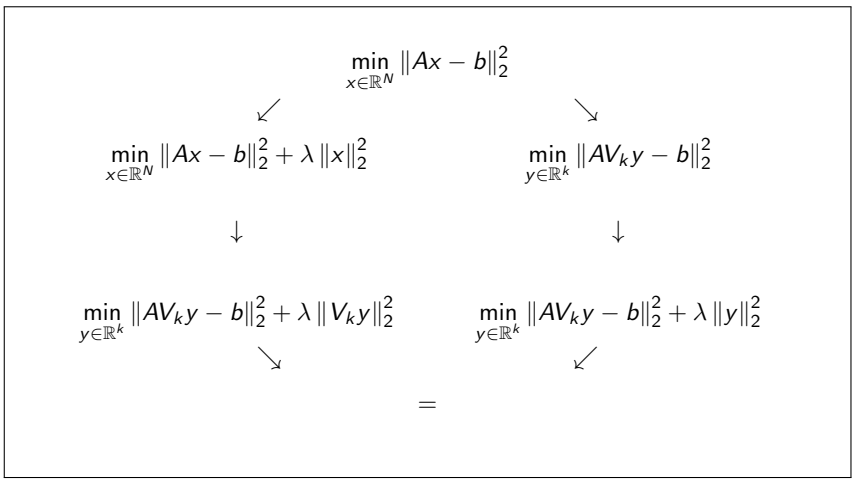


Figure 1.12: Sketch of the equivalence between the"first-regularize-then-project" and the "first-project-then-regularize" approaches. Inspired by a figure in [Han10, Chapter 6].

of x is unknown we are interested in finding a good statistical estimator for it. For example, if e is Gaussian white noise, then the solution of the least squares problem (1.6) is the maximum likelihood estimator of x.

The key purpose in Bayesian inference is to integrate results obtained from measurements with prior information. In this setting, left preconditioners for the linear system (1.2) (where the original problem is replaced by $M_lAx = M_lb$) are related to the distribution of the noise e, in the sense that M_l encodes information about the covariance of e. On the other hand, right preconditioners (where the original system is replaced by $AM_ry = b$ for $x = M_ry$), are related to previous available information about the solution, i.e., the prior for the random variable x. For this reason, in the framework of regularizing iterative methods, right preconditioners can be called prior-conditioners [Cal07]. In particular, we are interested in this latter kind of preconditioning. Note that, in general, the preconditioning matrices M_l and M_r are assumed to be invertible; otherwise, special care has to be taken to ensure that the preconditioned and the original system are equivalent.

When using a Krylov subspace method to solve a right-preconditioned system, the preconditioner M_r is embedded in the construction of the search space for the solution. This is particularly important in the setting of prior-conditioners, as the resulting Krylov subspace incorporates previous knowledge about the solution and therefore it becomes better suited for that particular application. For example, for preconditioned GMRES, we obtain a sequence of subspaces that, at iteration k, have the form

$$\mathcal{K}_k(AM_r, b) = \text{span}\{b, AM_rb, ..., (AM_r)^{k-1}b\}.$$

The subspaces in this sequence are nested in the sense that, at iteration k, $\mathcal{K}_{k-1} \subseteq \mathcal{K}_k$.

The regularization matrix appearing in Tikhonov problems in general form (1.11) can be naturally associated to right preconditioning. As already seen in Section 1.4, if we assume for simplicity that the regularization matrix L is square and non-singular, we can easily transform (1.11) into standard form, i.e., solve

$$\min_{\bar{x}\in\mathbb{R}^n}\{\|AL^{-1}\bar{x}-b\|_2^2+\lambda\|\bar{x}\|_2^2\} \quad \text{and then take} \quad x = L^{-1}\bar{x}. \tag{1.32}$$

Problem (1.32) is then equivalent to (1.11), but now the inverse of the regularization matrix acts as a right-preconditioner for the matrix A. If (1.32) is solved using a Krylov method,

as mentioned above, the effect of the regularization matrix L can be directly observed in the construction of the Kylov subspace for the solution through right preconditioning. In particular, at iteration k, projecting (1.32) into a Krylov subspace spanned by the columns of V_k leads to

$$\bar{y}_k = \arg\min_{\bar{y}\in\mathbb{R}^k} \{\|A \underbrace{L^{-1}V_k}_{Z_k} \bar{y} - b\|_2^2 + \lambda\|V_k\bar{y}\|_2^2\}, \quad \text{so that} \quad x = \underbrace{L^{-1}V_k}_{Z_k}\bar{y}_k. \tag{1.33}$$

A possible interpretation of (1.33) is that a solution of (1.32) in sought in the preconditioned subspace $\mathcal{R}(Z_k) = \mathcal{R}(L^{-1}V_k)$. In the case of GMRES, for example,

$$\mathcal{R}(Z_k) = L^{-1}\mathcal{K}_k(AL^{-1}, b) = \text{span}\{L^{-1}b, L^{-1}(AL^{-1})b, ..., L^{-1}(AL^{-1})^{k-1}b\}. \tag{1.34}$$

This approach is explored for GMRES in [HJ07], where a so-called smoothing norm regularization is incorporated into the approximation subspace for the solution through priorconditioning. It is interesting to note that, in [HJ07], no regularization term is considered, i.e., $\lambda = 0$ in (1.33), so smoothness in the solution is only encouraged through the construction of the space (1.34). In [GL19], included in Chapter 4, this method is called GMRES(L). Note that specific preconditioners can significatively improve the regularizing properties of GMRES, and therefore deliver better approximations for the solution.

A similar framework to the one described in (1.33) arises in the context of ℓ_2-ℓ_p regularization (1.16) solved by the IRLS or IRN methods. As previously explained in Section 1.4, the ℓ_2-ℓ_p regularization problem can be approximated by a sequence of reweighted Tikhonov problems of the form

$$x_k = \arg\min_{x\in\mathbb{R}^n} \{\|b - Ax\|_2^2 + \lambda\|W_k x\|_2^2\}. \tag{1.35}$$

The details of these derivations are unfolded in Chapter 3. If each of the problems of the form (1.35) approximating ℓ_2-ℓ_p is solved using an iterative solver, then the IRLS and IRN methods intrinsically rely on an inner-outer iteration scheme. This can be computationally expensive because a minimization problem is fully solved at each outer step, when the weights are updated. An alternative to avoid nested loops of iterations is to update the weights W_k as soon as a new approximation of x is available, i.e., at each iteration of an iterative solver for (1.35), leading to iteration-dependent preconditioning when (1.35) is transformed into standard form, i.e., we solve

$$\bar{x}_k = \arg\min_{\bar{x}\in\mathbb{R}^n} \{\|b - AW_k^{-1}\bar{x}\|_2^2 + \lambda\|\bar{x}\|_2^2\} \quad \text{and then take } x_k = W_k^{-1}\bar{x}_k. \tag{1.36}$$

Note that the regularization matrix W_k is square and invertible (this can be assumed through suitable thresholding [HLM$^+$17, WR08a]). This motivates the use of flexible Krylov methods.

We emphasize that, recalling (1.23), TV regularization can be formally presented in a very similar fashion to equation (1.35), where we have added a discrete representation of a gradient operator D:

$$x_k = \arg\min_{x\in\mathbb{R}^n} \{\|b - Ax\|_2^2 + \lambda\|W_k Dx\|_2^2\}. \tag{1.37}$$

However, since D is not invertible, TV regularization needs to be dealt with differently than ℓ_2-ℓ_p regularization; more details can be found in [GL19], in Chapter 4.

1.5.3 Flexible Krylov Methods

Flexible Krylov methods are modifications of standard Krylov methods that allow variable preconditioning. In particular, we are interested in flexible GMRES (FGMRES), derived from the flexible Arnoldi algorithm [SS86], and flexible LSQR (FLSQR), based on the flexible Golub-Kahan algorithm [CG19]. Given a sequence of square iteration-dependent preconditioners $\{W_k^{-1}\}_{k\geq 1}$, both FLSQR and FGMRES are based on the general partial decomposition of $A \in \mathbb{R}^{m\times n}$:

$$AZ_k = U_{k+1}\overline{G}_k, \quad \text{with} \quad \overline{G}_k \in \mathbb{R}^{(k+1)\times k}, \tag{1.38}$$

which is updated at each iteration k, and where $\overline{G}_k$ is upper Hessenberg, the columns of $U_{k+1} \in \mathbb{R}^{m\times(k+1)}$ are orthogonal and $u_1 = b/\|b\|$. For FGMRES, given a square matrix $A \in \mathbb{R}^{n\times n}$, $Z_k = [W_1^{-1}u_1, ..., W_k^{-1}u_k]$. Instead, for FLSQR, $Z_k = [(W_1^{-1})^2 v_1, ..., (W_k^{-1})^2 v_k]$, and the vectors v_i for $i = 1, ..., k$ are defined by updating the following partial factorization of A^T at each iteration k

$$A^T U_{k+1} = V_{k+1}S_{k+1}, \quad \text{with} \quad S_{k+1} \in \mathbb{R}^{(k+1)\times(k+1)}, \tag{1.39}$$

where S_{k+1} is upper triangular and V_{k+1} has orthogonal columns. More details on the flexible Arnoldi algorithm and the flexible Golub-Kahan process can be found in [GNL21], included in Chapter 3, along with pseudocodes for the updates of such algorithms.

An important feature of flexible Krylov subspaces is that, when used in combination with Tikhonov regularization, the "first-regularize-then-project" and "first-project-then-regularize" approaches are not equivalent anymore, even for a fixed regularization parameter λ. This fact is sketched in Figure 1.13.

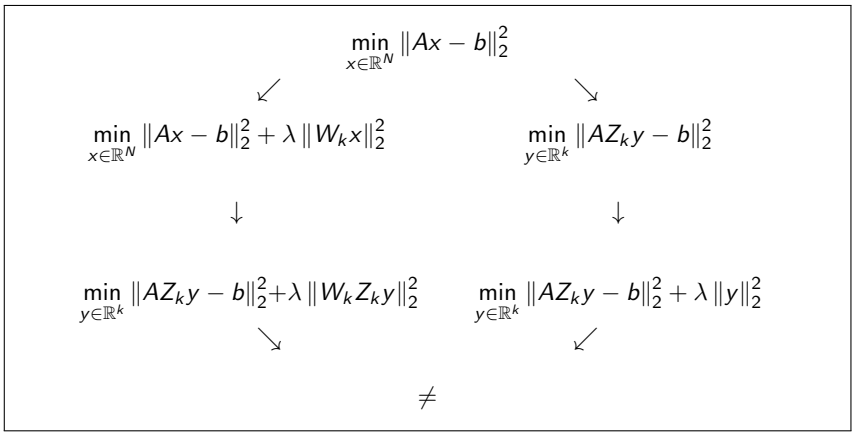


Figure 1.13: Sketch of the lack of an equivalence between the"first-regularize-then-project" and the "first-project-then-regularize" approaches in the case of flexible Krylov methods.

The general standard Krylov method update at iteration k defined in Algorithm 3 can be modified to describe a general framework for FLSQR and FGMRES and to be also used in combination with Tikhonov regularization. First, the flexible partial factorization (1.38) is updated (along with (1.39) for FLSQR). Second, we now have two different choices for the projection step. On one hand, if we adopt the "first-project-then-regularize" approach, for $x_k = Z_k y_k$,

$$y_k = \arg\min_{y\in\mathbb{R}^k} \{\|b - AZ_k y\|_2^2 + \lambda\|y\|_2^2\} = \arg\min_{y\in\mathbb{R}^k} \{\|\|b\|e_1 - \overline{G}_k y\|_2^2 + \lambda\|y\|_2^2\}. \tag{1.40}$$

This is used, for example, in [CG19] to compute sparse solutions of inverse problems using ℓ_2-ℓ_1 regularization. On the other hand, using the"first-regularize-then-project" approach

leads, for $x_k = Z_k y_k$, to a minimization problem of the form:

$$y_k = \arg\min_{y \in \mathbb{R}^k} \{\|b - AZ_k y\|_2^2 + \lambda \|W_k Z_k y\|_2^2\} = \arg\min_{y \in \mathbb{R}^k} \{\|\|b\| e_1 - \overline{G}_k y\|_2^2 + \lambda \|R_k y\|_2^2\}, \tag{1.41}$$

where $W_k Z_k = Q_k R_k$ is the QR decomposition of the tall and skinny matrix $W_k Z_k \in \mathbb{R}^{n \times k}$ that can be computed cheaply at each iteration if $k \ll n$ [GNL21]. The general update at iteration k for FLSQR and FGMES can be found in Algorithm 4.

Algorithm 4 Flexible Krylov method update for FLSQR and FGMRES at iteration k

1. Expand the Krylov subspace by adding a column z_k to the matrix Z_k and updating flexible decomposition (1.38) (involving also (1.39) if we are using FLSQR).
2. Solve a projected LS problem of the form (1.40) if following the "first-project-then-regularize" approach, or of the form (1.41) if following the "first-regularize-then-project" approach.
3. Compute the approximation $x_k = Z_k y_k$.

Flexible Krylov-Tikhonov methods using the "first-regularize-then-project" approach are exploited in [GNL21], in Chapter 3, to define two new algorithms (called IRW-FGMRES and IRW-FLSQR) for computing a solution to the smoothed version of the ℓ_2-ℓ_p regularization problem (1.16) (as described in Section 1.4). Moreover, the "first-regularize-then-project" approach is used in [GNL21] to prove the convergence of the solutions computed using the two new algorithms to a stationary point of the smooth approximation of the ℓ_2-ℓ_p regularization problem for $0 < p \leq 2$ (which is the unique minimizer when $p \geq 1$). As explained in Section 1.4, the smoothed version of ℓ_2-ℓ_p regularization can be approximated by a sequence of quadratic problems (1.35) in Tikhonov form (1.11). A possible interpretation of IRW-FGMRES and IRW-FLSQR is to consider that we are partially solving each problem in the sequence in a single solution (flexible Krylov) subspace of increasing dimension. Therefore, at each iteration k, the solution subspace is richer and bigger, and we expect (as supported by the numerical experiments in [GNL21]), that for a relatively small value of k we obtain solutions that approximately fully solve the kth problem in the sequence as defined in (1.35).

Other classical methods to compute a solution of (1.16), used in [GNL21] to perform comparisons with the new algorithms IRW-FGMRES and IRW-FLSQR, are the fast iterative shrinkage thresholding algorithm (FISTA) [BT09], and the sparse reconstruction by separable approximation algorithm (SpaRSA) [WNF09]. FISTA is an accelerated first-order optimization method, based on a forward-backward splitting approach where the minimization of the two terms in (1.16) is treated separately. This involves forward steps in the direction of the gradient of the fit-to-data term, which are alternated with implicit subgradient steps, or backward steps, involving the proximal mapping of the regularization term. SpaRSA is an algorithmic framework that relies on a quadratic separable approximation of parts of the objective function in (1.16).

Flexible Krylov methods are used in Chapter 4 to solve inverse problems imposing TV regularization. In particular, a new algorithm called TV-FGMRES is presented in [GL19]: this method features iteration-dependent preconditioning coming from a suitable transformation of a sequence of reweighted regularization terms in the form of (1.37). In [GL19], $\lambda = 0$ in (1.37) and the problem is solved at each iteration in a flexible Krylov subspace of increasing dimension. It is very interesting to see how this iteration-dependent preconditioning, that incorporates information about the piece-wise constant properties of

the solution, modifies the space where the solution is sought. In fact, [GL19, Section 3] presents an illustrative instance of this behavior for a simple one-dimensional deconvolution example. Without being exhaustive, here we list some classical methods to compute solutions of TV regularization (1.21). Fast first order iterations based on generalizations of the fast iterative shrinkage and thresholding algorithm (FISTA) are used in [BT09] (in [GL19], this method is called FBTV). Another strategy, proposed in [OBG$^+$05], consists in constructing a sequence of minimization problems in the general form (1.14), where the regularization term is $R(x) = \mathcal{D}(x, x_{k-1})$, x_{k-1} is the approximated solution computed for the previous problem in the sequence, and $\mathcal{D}(u, v)$ is the Bregman distance associated to the continuous TV functional. Fixed point iterations are used in [VO98] in combination with preconditioned conjugate gradient.

In the following chapters, the general theory of Krylov subspace methods for large-scale linear inverse problems presented in this introduction will be used to develop novel algorithms in combination with different variational regularization schemes.

Chapter 2

Krylov Methods for Inverse Problems: surveying classical, and introducing new, algorithmic approaches

This chapter surveys well-established regularizing projection methods based on Krylov subspace methods and introduces a new class of algorithms that leverages Tikhonov regularization and Krylov subspace methods in a principled way. This is a joint work with Silvia Gazzola, which is published in GAMM - Mitteilungen as open access publication [GL19].

2.1 Outline of the paper

This paper is set in the context of Krylov-Tikhonov methods for large-scale linear ill-posed problems described in Chapter 1. In particular, it draws special attention to the regularization parameter choice for the Tikhonov problem (1.11). After revising existing parameter choice rules, which usually involve (or can be re-written as) a minimization problem, Krylov-Tikhonov methods equipped with a parameter choice rule are presented as a bilevel optimization problem.

This paper has two very distinct parts. On one hand, as the name suggests, it surveys well-established regularizing projection methods based on Krylov subspace methods, in the style of Section 1.4. In particular, it centers around the methods based on the Golub-Kahan bidiagonalization algorithm introduced in Section 1.5 and Krylov-Tikhonov methods. On the other hand, it presents a new algorithmic approach for Krylov-Tikhonov regularization that provides a reliable regularization parameter choice strategy.

Since the new algorithmic framework for Krylov-Tikhonov regularization proposed in this paper involves the approximation of classical parameter choice criteria using Gaussian quadrature rules, the second part if this paper includes a short review on Gaussian quadrature rules and their well-established links with the Golub-Kahan bidiagonalization algorithm. Moreover, when the discrepancy principle is used as the parameter choice rule for the new algorithmic approach, we can prove that the computed solution converges to the Tikhonov solution with regularization parameter determined by the discrepancy

principle. Finally, a numerical example in image deblurring and a numerical example in computed tomography (as described in Section 1.2) are given to illustrate the performance and efficiency of this new method.

Note that most of the content included in the introduction of the paper [GL20, Section 1] has already been presented in Chapter 1 of this thesis, with some different details and citations, and can therefore be skipped by the reader. Chapter 1 also includes some of the material present in [GL20, Section 2].

The notation of Chapter 1 of this thesis is mainly coherent with the notation in the paper. However, some discrepancies are summarized in Table 2.1.

Table 2.1: Notational discrepancies between this thesis and paper [GL20]

Discrepancies	Thesis	Paper
Tikhonov regularization parameter	λ	α
Tikhonov solution	$x_{\lambda,L}^{Tik}$	$x(\alpha)$
Krylov subspace related to optimality condition	$\mathcal{K}_k(C,d)$ and $\mathcal{K}_k(C',d')$	$\mathcal{S}_k^{(1)}$ and $\mathcal{S}_k^{(2)}$

A list of typographical errors appeared in the published version of [GL19]. These are of two types: underscores that did not appeared as underscores and missing letters. Since the first kind does not affect the readability of the paper, we will just list the latter:

- Page 32 of this thesis (page 2 of [GL19]): in the last paragraph, '$(A) \cap (B)$' should be '$\mathcal{N}(A) \cap \mathcal{N}(B)$'.
- Page 33 of this thesis (page 3 of [GL19]): after the definition of $L_A^\dagger$, '(L)' should be '$\mathcal{N}(L)$'.
- Page 37 of of this thesis (page 7 of [GL19]): after equation (17), 'B_k and B_k' should be 'B_k and $\bar{B}_k$'.
- Page 39 of this thesis (page 9 of [GL19]): In the table, the expression for the UPRE parameter choice rule should be '$\frac{1}{m}\|Ax(\alpha)-b\|^2+2\frac{\eta^2}{m}\text{trace}(A(A^TA+\alpha I)^{-1}A^T)-\eta^2$'.
- Page 46 of this thesis (page 16 of [GL19]): before equation (48), 'convexity of $_k$' should be 'convexity of $\mathcal{L}_k$'.
- Page 47 of this thesis (page 17 of [GL19]): before equation (50), 'and $\{_k\}_k$' should be 'and $\{\mathcal{G}_k\}_k$'; in the last paragraph, 'At iteration k, $_k$ in (49)' should be 'At iteration k, $\mathcal{G}_k$ in (49)' and 'once the bound $_k$ is computed' should be 'once the bound $\mathcal{G}_k$ is computed'.
- Page 48 of this thesis (page 18 of [GL19]): in the third paragraph, 'lower bounds $_k$' should be 'lower bounds $\mathcal{G}_k$'.

2.2 Published paper

Statement of Authorship

This declaration concerns the article entitled:			
Krylov methods for inverse problems: Surveying classical, and introducing new, algorithmic approaches			
Publication status (tick one) **Draft manuscript** ☐ **Submitted** ☐ **In review** ☐ **Accepted** ☐ **Published** X			
Publication details (reference)	S. Gazzola and M. Sabaté Landman. Krylov methods for inverse problems: Surveying classical, and introducing new, algorithmic approaches. *GAMM-Mitteilungen*. 2020; e202000017. https://doi.org/10.1002/gamm.202000017		
Copyright status (tick the appropriate statement) I hold the copyright for this material X Copyright is retained by the publisher, but I have been given permission to replicate the material here ☐			
Candidate's contribution to the paper (provide details, and also indicate as a percentage)	The convergence proof has been mainly derived by the author of the thesis (80%) All authors contributed equally to the presentation of the content (50%) All authors contributed equally to the numerical experiments (50%)		
Statement from Candidate	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature.		
Signed	*Malena Sabaté Landman*	**Date**	10th Feb 2021

Last update: Feb 2019

Received: 2 January 2020 | Revised: 16 June 2020 | Accepted: 6 July 2020

DOI: 10.1002/gamm.202000017

ORIGINAL PAPER

Krylov methods for inverse problems: Surveying classical, and introducing new, algorithmic approaches

Silvia Gazzola | **Malena Sabaté Landman**

Department of Mathematical Sciences, University of Bath, Bath, UK

Correspondence
Silvia Gazzola, Department of Mathematical Sciences, University of Bath, Bath, UK. Email: S.Gazzola@bath.ac.uk

Funding information
Silvia Gazzola is partially supported by EPSRC, Grant/Award Number: EP/T001593/1; Malena Sabaté Landman is supported by a scholarship from the EPSRC Centre for Doctoral Training in Statistical Applied Mathematics at Bath (SAMBa), Grant/Award Number: EP/L015684/1

Abstract

Large-scale linear systems coming from suitable discretizations of linear inverse problems are challenging to solve. Indeed, since they are inherently ill-posed, appropriate regularization should be applied; since they are large-scale, well-established direct regularization methods (such as Tikhonov regularization) cannot often be straightforwardly employed, and iterative linear solvers should be exploited. Moreover, every regularization method crucially depends on the choice of one or more regularization parameters, which should be suitably tuned. The aim of this paper is twofold: (a) survey some well-established regularizing projection methods based on Krylov subspace methods (with a particular emphasis on methods based on the Golub-Kahan bidiagonalization algorithm), and the so-called hybrid approaches (which combine Tikhonov regularization and projection onto Krylov subspaces of increasing dimension); (b) introduce a new principled and adaptive algorithmic approach for regularization similar to specific instances of hybrid methods. In particular, the new strategy provides reliable parameter choice rules by leveraging the framework of bilevel optimization, and the links between Gauss quadrature and Golub-Kahan bidiagonalization. Numerical tests modeling inverse problems in imaging illustrate the performance of existing regularizing Krylov methods, and validate the new algorithms.

KEYWORDS

hybrid methods, imaging problems, Krylov subspace methods, large-scale linear inverse problems, regularization parameter choice rules, Tikhonov regularization

1 | INTRODUCTION

This paper considers linear, large-scale, discrete ill-posed problems of the form

$$Ax_{\text{true}} + e = b_{\text{true}} + e = b\,, \tag{1}$$

where the matrix $A \in \mathbb{R}^{m\times n}$ represents a forward mapping that is ill-conditioned with ill-determined rank (ie, the singular values of A decay and cluster at zero without an evident gap between two consecutive ones), $x_{\text{true}} \in \mathrm{R}^n$ is the desired solution, and $e \in \mathrm{R}^m$ is some unknown noise that affects the data $b \in \mathrm{R}^m$. Systems like (1) typically stem from the discretization of first-kind Fredholm integral equations, and model inverse problems arising in a variety of applications, such

as (just to name a few): computed tomography in Medicine and Industry, image deblurring in Astronomy and Biology, inverse scattering in Geophysics, and parameter identification (see [1-5] and the references therein for more details). In this paper the noise e is assumed to be Gaussian white; although this may appear like a limitation, strategies for whitening the noise or approximating actual noise (even realized from distributions other than Gaussian) with Gaussian white noise are available in the literature; see, for instance, [1, Chapter 3] and [6,7]. In general, the linear system $Ax = b$ might not be consistent, essentially because of the presence of noise in b. However, even the noise-free linear system $Ax_{\text{true}} = b_{\text{true}}$ may not be consistent, essentially because one should not assume a perfect agreement between the adopted (discretized) model and the true underlying physical model.

In a discrete setting, the properties of inverse problems are best analyzed considering the singular value decomposition (SVD) of A [8, Chapter 2], that is,

$$A = \hat{U}\hat{\Sigma}\hat{V}^T, \quad \text{for} \quad \hat{U} = [\hat{u}_i]_{i=1,\dots,m} \in \mathbb{R}^{m\times m}, \hat{V} = [\hat{v}_i]_{i=1,\dots,n} \in \mathbb{R}^{n\times n}, \text{ and } \hat{\Sigma} = \text{diag}(\hat{\sigma}_1, \dots, \hat{\sigma}_{\min\{m,n\}}) \in \mathbb{R}^{m\times n}. \quad (2)$$

Here the columns $\hat{u}_i$ and $\hat{v}_i$ of the square orthogonal matrices $\hat{U}$ and $\hat{V}$ are the left and right singular vectors of A, respectively, and the diagonal entries $\hat{\sigma}_1 \geq \dots \geq \hat{\sigma}_{\min\{m,n\}} \geq 0$ of the rectangular matrix $\hat{\Sigma}$ are its singular values. Depending on the decay of the singular values of A, (1) is regarded as severely ill-posed (exponential decay) or moderately and mildly ill-posed (power decay); see [9, Chapter 2]. Let $A^\dagger$ denote the Moore-Penrose pseudoinverse of A [8, Chapter 5]. Because of the ill-conditioning of the coefficient matrix A, and the noise in the right-hand side vector b, the (unregularized) solution $x = A^\dagger b$ to (1) is not guaranteed to be a meaningful approximation to x_{true}. Indeed, assuming that the discrete Picard condition [10] holds for (1) (ie, the spectral coefficients $\hat{u}_i^T b_{\text{true}}$ of b_{true} decay on average faster than $\hat{\sigma}_i$), and expressing x with respect to the SVD of A, leads to

$$x = \sum_{i=1}^{\min\{m,n\}} \frac{\hat{u}_i^T b}{\hat{\sigma}_i}\hat{v}_i = \sum_{i=1}^{\min\{m,n\}} \frac{\hat{u}_i^T b_{\text{true}}}{\hat{\sigma}_i}\hat{v}_i + \sum_{i=1}^{\min\{m,n\}} \frac{\hat{u}_i^T e}{\hat{\sigma}_i}\hat{v}_i, \quad (3)$$

where, on the right-hand-side, the singular components corresponding to the small singular values and the very oscillating right singular vectors in the second sum dominate and spoil x (since $|\hat{u}_i^T e|$ is roughly constant because e is Gaussian white noise); see [1, Chapter 4] and the references therein for more details.

In order to compute a good approximation to x_{true}, one should therefore regularize (1), that is, replace (1) with a problem closely related to it that is less sensitive to perturbations in the data. In the following, a generic regularized solution of (1), obtained by applying a regularized inverse $A\text{reg}\dagger$ to the noise-corrupted data b, is denoted by $x_{\text{reg}} = A_{\text{reg}}^\dagger b$. Although many approaches are possible to achieve regularization, this paper mainly focuses on strategies related to Tikhonov regularization method, which computes a regularized solution $x_{\text{reg}} = x(\alpha)$ as follows

$$x(\alpha) = \arg\min_{x\in\mathbb{R}^n} \{ \|Ax - b\|^2 + \alpha\|Lx\|^2 \}, \quad \text{with either} \quad \underbrace{L = I}_{\text{(standard form)}} \quad \text{or} \quad \underbrace{L \neq I}_{\text{(general form)}}, \quad (4)$$

and where $\|\cdot\|$ denotes the vector 2-norm. In the above formulation, the regularization parameter $\alpha \geq 0$ balances the effect of the fit-to-data term $\|Ax - b\|^2$ and the regularization term $\|Lx\|^2$, which is defined with respect of a regularization matrix $L \in \mathrm{R}^{q\times n}$. If $L \neq I$, typical choices for L are rescaled finite differences approximations of derivative operators. The choices of both α and L contribute to the success of Tikhonov regularization method (4). L should encode some available information about the solution, in such a way that unwanted behaviors of the regularized solution $x(\alpha)$ show up in $Lx(\alpha)$, whose magnitude is penalized: the bigger α, the more penalization, which is typically desirable if one is confident about the choice of L. Many strategies have been proposed to conveniently choose L; see, for instance, [11-14]. This paper will not focus on the choice of L, but rather on parameter choice rules for α. Note that (4) can be equivalently reformulated as

$$x(\alpha) = \arg\min_{x\in\mathbb{R}^n} \left\| \begin{bmatrix} A \\ \sqrt{\alpha}L \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|, \quad \text{or} \quad x_{\text{reg}} = x(\alpha) = (A^TA + \alpha L^TL)^{-1}A^Tb = A_{\text{reg}}^\dagger b, \quad (5)$$

where the latter assumes that the null spaces of A and L intersect trivially, that is, $(A)\cap(L) = \{0\}$. Thanks to (5), one can conclude that computing the Tikhonov-regularized solution $x(\alpha)$ essentially amounts to the solution of a least-squares problem (ie, the first problem in (5)). An analytical expression for $x(\alpha)$ is obtained by considering the associated system

of the normal equations (ie, the second equation in (5)). Therefore, at least in theory, any direct solver for least squares problems can be employed to compute $x(\alpha)$. In order to analyze the behavior of the solution and its dependence on α, the preferred choice is to employ the SVD of A (defined in (2)) if $L=I$, or the generalized SVD (GSVD) of (A,L) (see [8, Chapter 8]) if $L\neq I$. Indeed, when $L=I$, convenient expressions of the regularized solution $x(\alpha)$ and the discrepancy $b-Ax(\alpha)$ as functions of α can be obtained using the SVD of A as follows

$$x(\alpha)=\sum_{i=1}^{\min\{m,n\}}\phi_i(\alpha)\frac{\hat{u}_i^T b}{\hat{\sigma}_i}\hat{v}_i\ ,\quad b-Ax(\alpha)=\sum_{i=1}^{\min\{m,n\}}(1-\phi_i(\alpha))(\hat{u}_i^T b)\hat{u}_i+\sum_{i=\min\{m,n\}+1}^{m}(\hat{u}_i^T b)\hat{u}_i\ ,\ \text{where}\quad \phi_i(\alpha)=\frac{\hat{\sigma}_i^2}{\hat{\sigma}_i^2+\alpha}\leq 1\ . \tag{6}$$

Here the scalars $\phi_i(\alpha)$, $i=1,\ \dots\ ,\ \min\{m,n\}$, are the so-called filter factors that dampen the contributions to the regularized solution from the small singular values associated with the high-frequency components: the bigger the regularization parameter α, the more the amount of smoothing. Relations (6) become handy when applying some parameter choice rules. For context, another well-known spectral filtering method worth mentioning is the truncated SVD method [1, Chapter 4]. Also, relations (6), show that Tikhonov method can be regarded as a special spectral filtering method. A noteworthy property of Tikhonov regularization method is that every Tikhonov-regularized problem (4) can be equivalently transformed into standard form [15]

$$\bar{x}(\alpha)=\arg\min_{\bar{x}\in\mathbb{R}^q}\{\|\bar{A}\bar{x}-\bar{b}\|^2+\alpha\|\bar{x}\|^2\}\ ,\quad \text{where}\quad \begin{array}{rcl}\bar{A}&=&AL_A^\dagger\ ,\\ x_0&=&(A(I-L^\dagger L))^\dagger b\ ,\\ x(\alpha)&=&L_A^\dagger\bar{x}(\alpha)+x_0\ ,\\ \bar{b}&=&b-Ax_0\ .\end{array} \tag{7}$$

In the above equations, $L_A^\dagger$ is the A-weighted pseudoinverse of L, defined as

$$L_A^\dagger=[I-(A(I-L^\dagger L))^\dagger A]L^\dagger\in\mathbb{R}^{n\times q}\ ,$$

so that $x(\alpha)$ is the sum of a vector lying in range$(L_A^\dagger)$ and a vector x_0 lying in (L). Note that, if $q=n$ and $L\in\mathrm{R}^{n\times n}$ is nonsingular, then $L_A^\dagger=L^{-1}$ and $x_0=0$; if $q\geq n$ and $L\in\mathrm{R}^{q\times n}$ is full-rank, then $L_A^\dagger=L^\dagger$. Tikhonov regularization in standard form may be much more computationally convenient than its general form counterpart, since tasks like setting the regularization parameter may become simpler (see, for instance, [1, Chapter 6]). Thanks to (7), in the following, unless stated otherwise, only Tikhonov regularization in standard form will be formally considered (ie, $L=I$ in (4)), with the convention that, with some abuse of notations, $\bar{A}$, $\bar{x}$, and $\bar{b}$ in (7) may formally replace A, x, and b, respectively, in (4).

Although all the derivations explained above are useful in understanding the basics behind (Tikhonov) regularization, they are of little practical use when it comes to large-scale problems (like the ones naturally associated, for instance, to imaging problems). In general, in these cases, the matrix A may be only available as a sparse matrix or in the form of a function that efficiently computes the actions of A and (possibly) A^T on vectors (ie, matrix-vector products). In other words, only fully storing and explicitly accessing the matrix A may require massive memory, let alone the computational cost of solving (5), which is typically of the order of $O(n^3)$ (unless A (and L) have some special structure that is preserved when computing factorizations and can be exploited; see, for example, [16,17]). In all these situations, the only computationally viable approach to recover a solution of (1) is to apply an iterative linear solver that, as far as A is concerned, only requires matrix-vector products with A and (possibly) A^T (these are commonly referred to as "matrix-free" methods). Note that, for problems originally expressed in general Tikhonov form whose regularization matrix L does not have an exploitable structure, $L_A^\dagger$ must also be computed iteratively; this is the case especially for 2D or 3D problems, when, for example, L is a reweighted finite differences approximation of a derivative operator (see, eg, [18]). Although sometimes it may be more desirable to regularize (1) by applying an iterative solver to (5), in some situations just applying an iterative solver to (1) can lead to a regularized solution. Such linear solvers are commonly dubbed iterative regularization methods.

In the remaining part of the paper, only iterative regularization methods will be considered. More precisely, Section 2 surveys some well-established approaches to iterative regularization, which include stand-alone iterative Krylov solvers for (1) and methods that combine Tikhonov regularization and projection onto Krylov subspaces; a special emphasis will be given on methods based on the Golub-Kahan bidiagonalization algorithm. Sections 3 and 4 deal with practical regularization parameter choice rules that can be efficiently employed when combining Tikhonov regularization and Krylov

projection methods, also introducing a new principled class of algorithms where computations for expanding the approximation (projection) subspace for the solution and choosing the regularization parameter are interlaced. Section 5 displays some numerical results on test problems in image deblurring and computed tomography, where the performances of methods that combine Krylov solvers and Tikhonov regularization are compared. Section 6 presents concluding remarks.

2 | REGULARIZING KRYLOV PROJECTION METHODS

As mentioned at the end of the previous section, when dealing with large-scale problems (1), only iterative solvers can generally be employed. The most natural way of computing a regularized solution in this setting is by "early" termination of the iterations of a (convergent, regularizing) solver applied to (1); see, for instance, [1, Chapter 6] and the references therein. Let x_k denote the approximate solution at the kth iteration of a linear solver for (1). During the first iterations, that is, for k "small enough," x_k approaches the solution x_{true} of the noise-free system, and the relative reconstruction error

$$\text{RRE}(x_k) = \frac{\|x_k - x_{\text{true}}\|}{\|x_{\text{true}}\|} \tag{8}$$

decreases. However, as k increases after a certain optimal iteration $\tilde{k}$ (ie, when $\text{RRE}(x_{\tilde{k}})$ is minimal), $\text{RRE}(x_k)$ starts increasing as well and, indeed, x_k approaches the (unregularized and unwanted) solution $x = A^{\dagger}b$ (3). This phenomenon is referred to as "semiconvergence" (after [19, Chapter IV]). In this framework, the number of iterations clearly plays the role of a regularization parameter that must be carefully set according to one or more stopping criteria (acting in this situation as regularization parameter choice rules): performing $k \ll \tilde{k}$ iterations leads to over-regularized solutions, while performing $k \gg \tilde{k}$ iterations leads to under-regularized solutions.

Many well-known iterative solvers (such as Landweber method, Kaczmarz method, and many Krylov subspace methods) are known to semiconverge, and are considered regularization methods according to the following classical definition [5, Chapter 3]

$$\lim_{\|e\|\to 0} \sup_{\|b-b_{\text{true}}\|\le\|e\|} \|x_{\text{reg}} - x_{\text{true}}\| = 0\ , \text{ where } x_{\text{reg}} \text{ is determined according to a given regularization parameter choice rule}\ . \tag{9}$$

It should be highlighted that not all the parameter-choice rules in use today adhere to this equation; see Section 3 for more details. When employing a regularizing iterative solver, $x_{\text{reg}} = x_{\hat{k}}$, where $\hat{k}$ is chosen according to a specified stopping rule (acting as a regularization parameter choice rule).

In a discrete setting one can argue that this definition is not of practical use and, indeed, despite many iterative solvers satisfying (9), the most effective ones are those that, having a comparable computational cost per iteration, reach the optimal iteration $\tilde{k}$ quite "quickly". Moreover, the quality of the reconstructed solution at iteration $\tilde{k}$ matters. In the framework of this paper, where regularization in norms other than the vectorial 2-norm is not considered, this usually means that the quality of $x_{\hat{k}}$ is comparable to the quality of a x_{reg} that would have been achieved by applying a direct filtering method; see, for example [20]. Indeed, a possible way of exploring the regularizing properties of iterative methods is to formally reformulate them as filtering methods, where the filter factors depend on the iteration number k: this has been done for basic stationary iterative methods, such as Landweber, but also for some Krylov subspace methods, such as CGLS; see [21, Chapter 6] and the references therein.

In the following, only Krylov subspace methods will be considered as iterative solvers for (1) (or (4)). Krylov subspace methods belong to the class of iterative projection methods [22, Chapter 5], that is, at the kth iteration, and assuming (without loss of generality) a zero initial guess for the solution, an approximate solution x_k is computed by imposing that x_k belongs to a k-dimensional (approximation) subspace $\mathcal{S}_k^{(1)}$ and that the residual vector is orthogonal to a k-dimensional (constraint) subspace $\mathcal{S}_k^{(2)}$. Both the spaces $\mathcal{S}_k^{(1)}$ and $\mathcal{S}_k^{(2)}$ are typically expanded as the iterations proceed and, for Krylov subspace methods, they are both Krylov subspaces, that is, subspaces of the form

$$\mathcal{K}_k(G, c) = \text{span}\{c, Gc, \ \dots\ , G^{k-1}c\}\ , \tag{10}$$

where G is a a square matrix and c is a vector of coherent length; referring to (1) or (4), G and c are defined in terms of A and b. Although, in general, there is no guarantee that the dimension of $\mathcal{K}_k(G, c)$ is k, in the following it will be assumed

(without loss of generality) that this is the case. Indeed, the dimension of $\mathcal{K}_k(G,c)$ is k if and only if the grade of the vector c (ie, the degree of the minimal polynomial of c with respect to G [22, Chapter]) is greater than or equal to k. Krylov subspace methods for system (1) primarily differ for the choice of the two subspaces $\mathcal{S}_k^{(1)}$ and $\mathcal{S}_k^{(2)}$, but also for specific formulations adopted at implementation stage (methods that share the same $\mathcal{S}_k^{(1)}$ and $\mathcal{S}_k^{(2)}$ but are implemented differently are indeed mathematically equivalent). When $\mathcal{S}_k^{(2)} = A\mathcal{S}_k^{(1)} = A\mathcal{K}_k(G,c)$, Krylov subspace methods can be reformulated as constrained optimization methods computing

$$x_k = \arg\min_{x\in\mathcal{K}_k(G,c)} \|Ax-b\| \ ; \tag{11}$$

see, for example, [22, Chapter 5]. In the following, only Krylov methods that, when applied to (1), satisfy (11), will be taken into account. Formally, at iteration k, all Krylov subspace methods can be reformulated to update a partial factorization of the form

$$AW_k = Z_{k+1}\overline{R}_k \ , \tag{12}$$

where $W_k \in \mathbb{R}^{n\times k}$ has orthonormal columns that span the approximation subspace for the solution $\mathcal{K}_k(G,c)$, $Z_{k+1} \in \mathbb{R}^{m\times(k+1)}$ has orthonormal columns, and $\overline{R}_k \in \mathbb{R}^{(k+1)\times k}$. Along with the partial factorization (12), for some Krylov methods another factorization involving A^T is jointly updated at the kth iteration (this is, for instance, the case for LSQR method).

Using decomposition (12) and the properties of the matrices appearing therein, problem (11) can be reformulated as follows

$$\text{compute } x_k = W_k y_k, \text{ where } y_k = \arg\min_{y\in\mathbb{R}^k} \|AW_k y - b\| = \arg\min_{y\in\mathbb{R}^k} \|Z_{k+1}(\overline{R}_k y - Z_{k+1}^T b)\| = \arg\min_{y\in\mathbb{R}^k} \|\overline{R}_k y - Z_{k+1}^T b\| \ . \tag{13}$$

The rightmost problem appearing in (13) is a projected least squares problem of dimension $k \ll \min\{m,n\}$ and, therefore, when adopting Krylov subspace methods to approximate the solution of (1), the task of solving (regularizing) a full-dimensional system (1) is reduced to the task of solving a sequence of projected least squares problems of the form (13). Linking to the notations introduced in Section 1 to denote a generic regularization method, x_k in (11) and (13) can be expressed as $x_k = x_{\text{reg}} = A_{\text{reg}}^\dagger b$, where $A_{\text{reg}}^\dagger = W_k \overline{R}_k^\dagger Z_{k+1}^T b$ (note that x_{reg} depends on the number of iterations k, that acts as a regularization parameter). Table 1 lists some relevant Krylov subspace methods, giving some details about $\mathcal{K}_k(G,c)$ (equivalently, $\mathcal{S}_k^{(1)}$ and $\mathcal{S}_k^{(2)}$) and the matrices appearing in (12).

It must be stressed once more that, with some abuse of notation, here A is assumed to refer to the original coefficient matrix appearing in (1), or to the matrix $\overline{A} = AL_A^\dagger$ appearing in (7). In the latter case, the regularization matrix L affects the approximation subspace for the solution, in that $L_A^\dagger$ can be formally regarded as a right preconditioner for the linear system in (1). Indeed, when computing $x_k \in \mathcal{K}_k(G,c)$ according to (11), (13), and properly accounting for the transformation (7), the following expression for the kth approximate solution should be considered

$$x_k \in x_0 + \mathcal{K}_k(G,c) \ , \quad \text{where } x_0 \in \mathcal{N}(L), G \text{ is defined in terms of } A \text{ and } L_A^\dagger, c \text{ is defined in terms of } b, A, \text{ and } x_0 \ ; \tag{14}$$

TABLE 1 Examples of Krylov subspace methods commonly employed as iterative regularization methods

Method	Factorization (12)	$\mathcal{S}_k^{(1)}$	$\mathcal{S}_k^{(2)}$	W_k	Z_k	$\overline{R}_k$	References
GMRES	Arnoldi	$\mathcal{K}_k(A,b)$	$A\mathcal{K}_k(A,b)$	V_k	V_k	$\overline{H}_k$ Upper Hessenberg	[26]
RR-GMRES range-restricted GMRES	Arnoldi	$\mathcal{K}_k(A,Ab)$	$A\mathcal{K}_k(A,Ab)$	V_k	V_k	$\overline{H}_k$ Upper Hessenberg	[27]
LSQR (equivalent to CGLS)	Golub-Kahan bigiagonalization (GKB)	$\mathcal{K}_k(A^TA,A^Tb)$	$A\mathcal{K}_k(A^TA,A^Tb)$	V_k	U_k	$\overline{B}_k$ Lower bidiagonal	[28,29]

see [1, Chapter 8; 23]. However, $L_A^\dagger$ should not be practically regarded as traditional preconditioner: indeed, it does not contribute to speeding up the convergence of the solver (which is undesirable because of semiconvergence [24]) but, on the contrary, most often results in a delayed semiconvergence of the iterative regularization method to a regularized solution of enhanced quality. Indeed, incorporating $L \neq I$ within the Krylov subspace (10) can largely contribute to building a better approximation subspace for the solution of (1) (and (4)). In other words, similarly to Tikhonov regularization method in general form (4) and to the so-called "truncated" GSVD method [25], including $L_A^\dagger$ in (14) enforces regularization in the (semi)norm $\|L \cdot\|$ (even if no penalization appears in (11), (13)).

Much recent (and ongoing) research is studying regularizing properties of Krylov methods of the form (13), expanding on the theoretical definition (9). For instance, [30-32] investigate the quality of the approximation of the SVD of A obtained when considering the SVD of the projected matrix $\overline{R}_k$; [33] investigates whether properties like the discrete Picard condition are inherited by the projected problem in (13). The results of these studies often depend on the original problem (1) being severely, moderately, and mildly ill-posed (the first category usually enjoying better regularization properties). Also, more thoughtful investigations of the effect of the noise on the approximation subspace (10) for the solution have been performed in [34-36] for a variety of Krylov subspaces, concluding that, in some cases, an accurate estimate of the amount of noise in the data b can be also recovered. Thanks to these new results, new insight into the semiconvergence phenomenon for Krylov methods of the form (13) can be gained: namely, these methods semiconverge because of a combination of the noise entering the approximation subspace for the solution $\mathcal{K}_k$, and the projected matrix $\overline{R}_k$ inheriting the ill-conditioning of A as k increases.

The methods based on the Arnoldi algorithm require only one matrix-vector product with A per iteration; therefore, assuming that the dominant cost per iteration of an iterative solver is the computation of matrix-vector products with matrices related to A (which might not be the case for a sparse or structured A), these methods are the cheapest ones among the Krylov solvers listed in Table 1. However, the use of GMRES as a regularization method is not widespread. Indeed, although some results on the SVD approximation properties of GMRES have been recently obtained (even in a continuous setting [37]), in some situations (eg, when A is highly non-normal), GMRES performs worse than other Krylov subspace methods. One of the reasons for the bad performance of GMRES as an iterative regularization method may be that the approximation subspace for the solution explicitly contains the vector b, which is noise-corrupted (indeed, this prompted the introduction of RR-GMRES [26]). Moreover, the approximation subspaces generated by both GMRES and RR-GMRES may not be suitable to represent the exact solution as they basically only contain information about the range of A, or may be affected by a severe 'mixing' of the singular value components of the solution (so that GMRES and RR-GMRES cannot be interpreted as filtering methods); see [38]. When A is symmetric, GMRES (resp. RR-GMRES) reduces to the well-known MINRES [39] (resp. MR-II [40]) method, whose regularization properties are well-studied and well-accepted; see [41]. It should be stressed that, even if the original coefficient matrix A in (1) is square, if $L \neq I$ is rectangular, then the preconditioned matrix $\overline{A} = AL_{\mathrm{A}}^\dagger$ in (14) is not square anymore: in this situation, applying GMRES (and variants thereof) might still be possible by either defining square counterparts of such rectangular matrices (see [42,43]) or by revisiting the standard form transformation (7) (see [44]). When a regularization matrix is incorporated, GMRES applied to the transformed system may perform significantly better than its standard counterpart [45].

When it comes to regularization, LSQR is arguably the most widespread Krylov method among the ones listed in Table 1; in the remaining part of this paper, unless otherwise stated, only LSQR will be considered as a Krylov projection method. More details about LSQR are hereby provided. Given the initial vectors $v_0 = 0$ and $u_1 = b/\|b\|$, and taking $\zeta_1 = \|b\|$, the jth iteration of the Golub-Kahan bidiagonalization (GKB) algorithm computes

$$\rho_j v_j = A^T u_j - \zeta_j v_{j-1}, \quad \zeta_{j+1} u_{j+1} = A v_j - \rho_j u_j, \quad j = 1, 2, \dots, \tag{15}$$

with ρ_j and ζ_{j+1} chosen so that $\|v_j\|$=1 and $\|u_j\|$=1, respectively. Breakdown of GKB happens if $\rho_j = 0$ (ie, range(U_k) is invariant with respect to AA^T) or if $\zeta_{j+1} = 0$ (ie, range(V_k) is invariant with respect to A^TA): this can only happen if (1) is consistent (which is not necessarily the case considered in this paper) or if x_k exactly solves the least squares problem $\min x \in \mathbb{R}^n \|Ax - b\|$ associated with (1) (which is not a desirable situation when performing regularization), respectively; see, for instance, [46] for more details. Otherwise GKB continues until the maximum dimensionality of the problem (ie, $\min\{m, n\}$) is reached. The

Assumption: The GKB algorithm does not breakdown, (16)

will be made from now on, without loss of generality. Indeed, as already emphasized, LSQR (and, in general, all the regularizing Krylov methods), are meaningful if the total number of iterations k is quite low (ie, $k \ll \min\{m, n\}$), so that one can usually safely assume that no breakdown will happen. Looking at (15) one can see that each iteration requires computing one matrix-vector product with A and one matrix-vector product with A^T. Equivalently, after k iterations of GKB are performed, one can write partial matrix factorizations of the form

$$AV_k = U_k B_k + \zeta_{k+1} u_{k+1} e_k^T = U_{k+1} \overline{B}_k \,, \qquad A^T U_k = V_k B_k^T \,, \tag{17}$$

where $V_k = [v_1, \ldots, v_k] \in \mathbb{R}^{n\times k}$ and $U_{k+1} = [U_k, u_{k+1}] = [u_1, \ldots, u_k, u_{k+1}] \in \mathbb{R}^{m\times(k+1)}$, are matrices whose orthonormal columns span the Krylov subspaces $\mathcal{K}_k(A^TA, A^Tb)$ and $\mathcal{K}_k(AA^T, b)$, respectively; B_k and B_k are lower bidiagonal matrices of the form

$$B_k = \begin{bmatrix} \rho_1 & & & & \\ \zeta_2 & \rho_2 & & & \\ & \ddots & \ddots & & \\ & & \zeta_{k-1} & \rho_{k-1} & \\ & & & \zeta_k & \rho_k \end{bmatrix} \in \mathbb{R}^{k\times k}, \quad \overline{B}_k = \begin{bmatrix} B_k \\ \zeta_{k+1} e_k^T \end{bmatrix} \in \mathbb{R}^{(k+1)\times k}. \tag{18}$$

Here and in the following, e_i denotes the ith canonical basis vector of $\mathbb{R}^d$, $d \geq i$. Note that the first decomposition appearing in (17) is a specific realization of the more general decomposition in (12).

Combining Tikhonov regularization and Krylov methods

Although, by applying a regularizing iterative solver to (1) and stopping the iterations sufficiently early, thanks to semiconvergence one is successful in computing a regularized solution to (1), special attention should be made to select a proper stopping criterion for the iterations. As hinted at the end of Section 1, an alternative approach to apply iterative regularization to (1), which does not rely on semiconvergence, is to apply (till convergence) an iterative solver to the Tikhonov-regularized problem (4) (most often considering its equivalent formulation as a damped least squares problem in (5)). By doing so, if the selected iterative solver is a projection method and, in particular, a Krylov subspace method, one is within the framework of the so-called *first-regularize-then-project* methods; see [1, Chapter 6]. In other words, by exploiting decomposition (12) and the properties of the matrices appearing therein, the kth iteration of these methods computes

$$x_k(\alpha) = \arg\min_{x\in\mathcal{K}_k(G,c)} \{\|Ax - b\|^2 + \alpha\|x\|^2\} = W_k y_k(\alpha) \,, \text{ where } y_k(\alpha) = \arg\min_{y\in\mathbb{R}^k} \{\|\overline{R}_k y - Z_{k+1}^T b\|^2 + \alpha\|y\|^2\}. \tag{19}$$

An upside of this approach is that the stopping criterion for the iterations plays a less crucial role in the regularization process. However, a clear downside is that, if a good value for the regularization parameter α is not available a priori (as it is often the case), then one should repeatedly fully solve problems of the form (4), one for every considered value of the regularization parameter (these may belong to a discrete set of values chosen a priori, or may be sequentially set by trying to satisfy some well-known parameter choice rules; see Section 3 for more details). Fortunately, when using a Krylov method to solve problem (4) in standard form for a varying α, most of the computations performed to determine $x_k(\alpha_\ell)$ for a given α_ℓ can be smartly rearranged or recycled to compute $x_k(\alpha_{\ell+1})$ when $\alpha_\ell \neq \alpha_{\ell+1}$ (exploiting, essentially, the so-called shift-invariance of Krylov subspaces, that is, $\mathcal{K}_k(G + \alpha_\ell I, c) = \mathcal{K}_k(G + \alpha_{\ell+1} I, c)$; see [47,48]). Note that, in general, the number of iterations k for solving (4) depends on α.

Another strategy to combine a Krylov (projection) method and Tikhonov method to compute a regularized solution of (1) is to perform a so-called *hybrid method*; see [4] and the references therein. At the kth iteration of a hybrid method, one projects problem (4) onto a Krylov subspace $\mathcal{K}_k$ (so that a formulation like the one in (13) is recovered), and then applies some (standard form) Tikhonov regularization to the projected least squares problem. By doing so, one is within the framework of the so-called *first-project-then-regularize* methods; see [1, Chapter 6]. In other words,

by using decomposition (12) and the properties of the matrices appearing therein, the kth iteration of these methods computes

$$x_k(\alpha_k) = \arg\min_{x\in\mathcal{K}_k(G,c)} \{ \|Ax-b\|^2 + \alpha_k\|x\|^2 \} = W_k y_k(\alpha_k) \,, \text{ where } y_k(\alpha_k) = \arg\min_{y\in\mathbb{R}^k} \{ \|\overline{R}_k y - Z_{k+1}^T b\|^2 + \alpha_k\|y\|^2 \} \,. \quad (20)$$

By comparing the two formulations (19) and (20) it is evident that, if $\alpha_k = \alpha$, then these two approaches are equivalent. When adopting hybrid methods (20), the search for a good regularization parameter is done on the fly using well-known parameter choice rules on the projected problem only, which has relatively small dimensions (if $k \ll \min\{m,n\}$) and is therefore less computationally demanding than working with the original large-scale problem (4); see [49] and Section 3 for more details. It should also be emphasized that, although α_k is a suitable regularization parameter for the kth iteration (ie, it is in principle a local choice, good for the kth projected Tikhonov problem (20) only), when k increases α_k seems to stabilize around a value that is good for the full-dimensional problem, too; see [50]. In the following, among the approaches (19) and (20), only the latter will be considered (unless stated otherwise).

When combining projection methods and Tikhonov regularization, one is effectively dealing with two-parameter methods, meaning that a good regularized solution to (1) is achieved by a sensible interplay of the number of performed iterations and the chosen Tikhonov regularization parameter. In particular, when comparing these methods to purely iterative Krylov methods (13), setting a good stopping rule for k is less crucial for recovering a suitable regularized solution, and often a larger approximation subspace $\mathcal{K}_k$ can be generated, where the regularized solution can be more accurately recovered.

Linking to the notations introduced in Section 1 to denote a generic regularization method, $x_k(\alpha_k)$ in (20) can be expressed as $x_{\text{reg}} = A_{\text{reg}}^{\dagger}$, where $A_{\text{reg}}^{\dagger} = W_k(\overline{R}_k^T\overline{R}_k + \alpha_k I)^{-1}\overline{R}_k^T Z_{k+1}^T b$ (note that x_{reg} depends on both the number of iterations and the Tikhonov parameter α_k).

3 | EXISTING PARAMETER CHOICE RULES FOR LARGE-SCALE PROBLEMS

Choosing one (or more) suitable regularization parameter(s) is a nontrivial and heavily problem-dependent aspect of every regularization method. For large-scale problems, this has been a widely-studied topic covered in the classical bibliography about Tikhonov, (purely) iterative, and hybrid regularization methods (see, eg, [21, Chapter 7], [51, Chapter 15]), as well as in more or less recent survey papers (see, eg, [49,52,53]). In the hybrid framework, two similar but differently principled approaches can be used: using a projected parameter choice scheme, or using an approximation of the full dimensional parameter choice scheme that only relies on information available within the projected problem. In this section, only a few parameter choice strategies that are well-established in these settings will be reviewed: they can all be formulated as minimization problems, whose objective functions $P(x_{\text{reg}})$ depend on one (or more) regularization parameter(s) through x_{reg} (ie, the regularized solution that is obtained by (4), or (13), or (20)). Table 2 gives more details about the strategies considered in this paper, specified for the Tikhonov regularization case (4), that is, $x_{\text{reg}} = x(\alpha)$; the information reported there (and in this section) is by no means exhaustive.

If a good estimate of the norm of the error in the right hand side $\|e\|$ is available, coming from existing knowledge about the problem or recovered by noise estimation algorithms (which may be the same Krylov solver employed for regularization [34,35]), the most widespread and reliable method is the so-called discrepancy principle [54]. It consists in choosing the regularization parameter such that the solution x_{reg} satisfies

$$\varepsilon := \tau\|e\| = \|Ax_{\text{reg}} - b\| \,, \quad (21)$$

where τ is a (safety) parameter, usually chosen slightly bigger than 1. When e is Gaussian white noise with covariance matrix $\eta^2 I$, one typically takes $\|e\| = \eta\sqrt{m}$ (ie, the expected value of $\|e\|$). When considering Tikhonov method (4), one applies a zero finder, for example, [55], to solve with respect to α the nonlinear equation $\varepsilon = \|Ax(\alpha) - b\|$. Note that this is an equivalent formulation to the (unconventional) one reported in Table 2; more details will be given in Section 4.2. When considering purely iterative methods satisfying (13), this corresponds to stopping the iterations as soon as the discrepancy norm $\|Ax_{\hat{k}} - b\|$ lies below ε for an iteration $\hat{k}$ (the subsequent iterations are guaranteed to have discrepancy norms lying below ε because of the optimality property of the residual in (13) that, in this case, coincides with the discrepancy). When

TABLE 2 Parameter choice rules considered in this paper, which can be (re)formulated as minimization problems min $P(x_{\text{reg}})$ over the set of admissible regularization parameters. The objective functions $P(x_{\text{reg}})$ reported in the second column are specified for Tikhonov regularization (4), that is, $x_{\text{reg}} = x(\alpha)$; notation-wise, $r(\alpha) = b - Ax(\alpha)$. The reported references contain implementation details about specific parameter rules, tailored to the considered regularization method

		References		
Parameter choice rule	**Objective function $P(x_{\text{reg}}) = P(x(\alpha))$**	**Tikhonov (4)**	**Krylov (13)**	**Hybrid Krylov (20)**
Discrepancy principle (DP)	$b^T((\alpha I - \alpha\bar{\varepsilon}^2 I + 2AA^T \log((AA^T + \alpha I)^{-1}))b - b^T(AA^T)^2 x(\alpha)$ (where $\bar{\varepsilon} \simeq \|e\|/\|b\|$)	[54,55]	[40]	[56,57]
Unbiased Predictive (UPRE)	$\frac{1}{}\|Ax(\alpha) - b\|^2 + 2\frac{\eta^2}{}$ trace$(A(A^TA + \alpha I)^{-1}A^T) - \eta^2$	[2, Chapter 7]	[2, Chapter 7]	[58]
Generalized Cross Validation (GCV)	$\frac{\|r(\alpha)\|^2}{(\text{trace}(I - A(A^TA+\alpha I)^{-1}A^T))^2}$	[59,60]	[21, Chapter 7]	[30,58,61]
L-curve	$\frac{(\log(\|r(\alpha)\|))'(\log(\|x(\alpha)\|))'' - (\log(\|r(\alpha)\|))''(\log(\|x(\alpha)\|))'}{(((\log(r(\alpha)))')^2 + (\log(x(\alpha))')^2)^{3/2}}$	[62]	[62]	[31,63]
Regińska criterion	$\|b - Ax(\alpha)\|\|x(\alpha)\|^\xi, \quad \xi > 0$	[64]	[52]	[31]

considering hybrid methods (20), at the kth iteration, one typically solves (with respect to α) the projected version of (21), that is,

$$\varepsilon = \|Ax_k(\alpha) - b\| = \|Ax_k(\alpha_k) - b\| = \|\overline{R}_k y_k(\alpha_k) - Z_{k+1}^T b\| \,. \tag{22}$$

The last equality in (22) holds thanks to the generic decomposition (12) and the properties of the matrices appearing therein. In other words, one typically determines the Tikhonov parameter α_k to be employed at the kth hybrid iteration by imposing (22), and sets a(n unrelated) stopping criterion for the iterations. For instance, the authors of [57] first determine a suitable iteration k assuring that equation (22) has a unique solution, apply a zero finder to solve (22), and then compute $x_k(\alpha_k)$. However the value of k so determined may be too small to guarantee a good approximation, and usually a few extra iterations are performed to improve the quality of the solution by using a richer approximation subspace. The authors of [56] derive the so-called "secant update" method, which solves (22) by performing one step of a secant-like zero finder at each hybrid iteration (to update the regularization parameter for the projected problem (20)), stopping when (22) is satisfied. As an upside, when a method (13) and (20) is adopted with the discrepancy principle, one can typically prove that the resulting strategy is a regularization method according to the definition given in (9). As downsides, the discrepancy principle relies on having an accurate estimate of $\|e\|$, and tends to oversmooth the regularized solution [21, Chapter 7].

Another parameter choice rule that can be employed when the standard deviation η of the noise e is known is the unbiased predictive risk estimator (UPRE), which is based on a statistical estimator of the mean squared norm of the predictive error, and consists in computing

$$\min U_{\text{reg}}(x_{\text{reg}}) \,, \quad \text{where} \quad U_{\text{reg}}(x_{\text{reg}}) = \frac{1}{m}\|Ax_{\text{reg}} - b\|^2 + \frac{2\eta^2}{m}\text{trace}(AA^\dagger_{\text{reg}}) - \eta^2 \,, \tag{23}$$

where the minimization happens over the set of admissible regularization parameters. When considering Tikhonov method (4) one typically applies a local optimization method (eg, Newton's method) to find a local minimizer of $U_{\text{reg}}(x_{\text{reg}}) = U_\alpha(x(\alpha))$ with respect to α; some strategies based on Krylov subspace methods, or randomized trace estimators, may be employed to efficiently estimate the trace term in (23) when adopting a *first-regularize-then-project* approach (19); see [49,65]. When considering purely iterative regularization methods (13) one should monitor the value of the functional $U_{\text{reg}}(x_{\text{reg}}) = U_k(x_k)$ at the kth iteration, and stop as soon as a significant growth happens. UPRE has been adapted in [58] to work in the hybrid framework (20), where a projected version of the optimization problem (23) is solved with respect to the Tikhonov parameter at each iteration k, to determine a suitable $\alpha = \alpha_k$ to be employed in (20).

Parameter choice rules that do not use $\|e\|$ nor any other information about e are sometimes called "heuristic methods" because it has been shown that no regularization method equipped with any of these rules can be proven to satisfy the

definition (9); see [5, Chapter 4] and [21, Chapter 7]. A popular strategy belonging to this class of rules is the so-called generalized cross validation (GCV), which is based on a statistical tool (cross validation) used to predict possible missing data values (so that the regularization parameter is chosen to obtain the best prediction with respect to any supposedly missing entry b_i, $i = 1, \dots, m$, of the vector b). In practice, GCV requires the computation of

$$\min G_{\text{reg}}(x_{\text{reg}}) , \quad \text{where} \quad G_{\text{reg}}(x_{\text{reg}}) = \frac{\|Ax_{\text{reg}} - b\|^2}{(\text{trace}(I - AA^{\dagger}_{\text{reg}}))^2} \tag{24}$$

and where the minimization happens over the set of admissible regularization parameters. This strategy was first proposed and analyzed in [59] for Tikhonov method (4). Application of GCV to purely iterative methods (13) is not straightforward, because of the different approaches that can be adopted to compute the trace at the denominator of G_{reg} in (24); see [[21], Chapter 7]. When adopting a *first-regularize-then-project* approach (19) in connection with large-scale (and possibly matrix-free) problems, and similarly to UPRE (23), the main difficulty in applying GCV is the computation of the denominator of the the GCV functional G_{reg}. This issue is commonly alleviated either by using a Monte Carlo approach [66] or by using stochastic trace estimators, for example, Hutchinson's stochastic trace estimator [65]. More precisely, [60] uses the latter strategy, by also exploiting links between GKB and Gaussian quadrature rules to approximate the GCV functional G_{reg} as a function of the Tikhonov regularization parameter α; once these approximations give bounds that are tight enough, they are used to determine a suitable value for α, and method (19) can be employed to find a regularized solution; see also [51, Chapter 15] and references therein for more details. GCV and variations thereof have also been used in connection with hybrid methods (20), based either on GKB or other decompositions of the kind (12): for instance, [61] adopts a weighted GCV approach, and [30,58] derive a modified GCV approach, where also insight about the relationship between a projected functional approximating the GCV functional $G_{\text{reg}}(x_{\text{reg}})$ and $G_{\text{reg}}(x_{\text{reg}})$ itself is given. All these hybrid strategies have in common that an approximated version of the optimization problem (24) is solved with respect to the Tikhonov parameter α at each iteration k, to determine a suitable $\alpha = \alpha_k$ to be employed in (20): in doing so, these strategies often overcome another known drawback of the GCV rule (24), namely, the fact that the GCV function $G_{\text{reg}}(x_{\text{reg}})$ is usually very flat around its minimum [67].

Another very popular scheme that does not require information about the noise e in (1) is the so-called L-curve criterion [62]: it consists in plotting the curve $\mathbb{L}(x_{\text{reg}}) = (\log(\|x_{\text{reg}}\|), \log(\|Ax_{\text{reg}} - b\|))$ with respect to different values of the regularization parameter (so that each point on the L-curve corresponds to one regularization parameter), and then finding the corner of such curve (which is hopefully shaped as an "L"). The rationale behind the L-curve criterion is that the corner of the L-curve is supposed to represent an good balance between the value of $\|x_{\text{reg}}\|$ (which should not be neither too small nor too big, to avoid over- or under-regularization, respectively) and the value of $\|Ax_{\text{reg}} - b\|$ (which should not be neither too small nor too big, to avoid under- or over-regularization, respectively). In practice, the L-curve criterion requires the computation of

$$\min(-\chi(\mathbb{L}(x_{\text{reg}}))) , \quad \text{where } \chi \text{ denotes the curvature of the L-curve,} \tag{25}$$

and where the minimization happens over the set of admissible regularization parameters. When considering Tikhonov method (4), any optimization routine can be run to solve (25) with $x_{\text{reg}} = x(\alpha)$ with respect to the Tikhonov regularization parameter α; see [62]. Although the L-curve criterion works well in many cases, when paired with Tikhonov method it might not be consistent with the classical definition of regularization method (9); see, for example, [68,69] and [[21], Chapter 7]. When considering a purely iterative method (13), the L-curve criterion can be meaningful only if both $\|Ax_k - b\|$ and $\|x_k\|$ are monotonic; while the former is guaranteed by the optimality property of the residual (11), the latter may not hold for all solvers (eg, among the ones listed in Table 1, GMRES and RR-GMRES). When dealing with large-scale problems by adopting a *first-regularize-then-project* approach, since one should typically solve one regularized problem for each of the points lying on the L-curve, a possible computationally feasible fix is to compute only a few points of the curve using the strategy (19), and then find the corner using interpolation techniques; see [62]. When adopting a hybrid method (20), a projected L-curve is computed at the kth iteration (ie, $\mathbb{L}_k(x_k(\alpha_k)) = (\log(\|x_k(\alpha_k)\|), \log(\|Ax_k(\alpha_k) - b\|))$) and an optimization problem analogous to (25) is solved with respect to the Tikhonov parameter α_k at each iteration k: computing every point on the projected L-curve is feasible (either because $k \ll \min\{m, n\}$ [31] or because computations for a given set of predetermined values of the Tikhonov parameter are simultaneously updated from one iteration to the next one [63]). When hybrid methods based on GKB are employed, the links between GKB and Gaussian quadrature rules have

been exploited in [70,71] to derive upper and lower bounds (ie, boxes) for the points on the L-curve (this is the so-called "L-ribbon" method) or to compute curvature bounds [72] at each iteration.

Closely related to the L-curve criterion is the Regińska criterion [64], which consists in solving

$$\min R_{\text{reg}}(x_{\text{reg}}) , \quad \text{where} \quad R_{\text{reg}}(x_{\text{reg}}) = \|Ax_{\text{reg}} - b\| \|x_{\text{reg}}\|^{\xi}, \quad \text{where} \quad \xi > 0 \quad \text{is fixed} , \tag{26}$$

and where the minimization happens over the set of admissible regularization parameters. This strategy was first proposed and analyzed in [64] for Tikhonov method (4); an algorithm to find a local minimum of $R_{\text{reg}}(x_{\text{reg}}) = R_\alpha(x(\alpha))$ with respect to α is derived in [73]. Application in the setting of purely iterative methods (13) was described in [52]. The Regińska criterion (26) has then been adapted in [31] to work within the framework of hybrid methods (20): essentially, this involves computing a projected Regińska functional at the kth iteration (ie, $R_{\text{reg}}(x_{\text{reg}}) = R_k(x_k(\alpha_k)) = \|Ax_k(\alpha_k) - b\| \|x_k(\alpha_k)\|^{\xi}$) and minimizing it with respect to the Tikhonov parameter α_k.

4 | A NEW BI-LEVEL OPTIMIZATION FRAMEWORK FOR REGULARIZATION

All the rules presented in Section 3 for hybrid regularization (ie, *first-project-then-regularize*) methods (20) have in common that, together with the regularization method itself, can be reformulated as a sequence of problems of the form

$$\min_{\alpha_k \geq 0} P_k(x_k(\alpha_k)) \quad \text{subject to} \quad x_k(\alpha_k) = \arg\min_{x \in \mathcal{K}_k} F(x, \alpha_k) , \quad k = 1, 2, \ldots \tag{27}$$

where $F(x, \alpha_k)$ is the objective function appearing in the first optimization problem in (20) and $P_k(x_k(\alpha_k))$ is a condition that allows a suitable choice of $\alpha_k \geq 0$ (typically a projected version of the strategies listed in Section 3 and also listed in Table 2). For instance, when using Regińska criterion,

$$P_k(x_k(\alpha_k)) = R_k(x_k(\alpha_k)) = \|Ax_k(\alpha_k) - b\| \|x_k(\alpha_k)\|^{\xi} = \|\overline{R}_k y_k(\alpha_k) - Z_{k+1}^T b\| \|y_k(\alpha_k)\|^{\xi} , \quad \xi > 0 , \tag{28}$$

where the functional $R_k(x_k(\alpha_k))$ can be regarded as a projected version of the functional R_{reg} appearing in (26), specific of hybrid methods (obtained by exploiting decomposition (12) and the properties of the matrices appearing therein). Problem (27) is formally a sequence of bi-level optimization problems, each consisting of a lower-level optimization problem whose solution $x_k(\alpha_k)$ is an argument of the higher-level minimization problem; see [74]. Thanks to the particular form of $F(x, \alpha_k)$ and assuming $k \ll \min\{m, n\}$, one can derive a closed-form solution for $x_k(\alpha_k)$, and substitute its expression in $P_k(x_k(\alpha_k))$, so that each problem (27) can be essentially regarded as a single-level optimization problem. Even in this favourable case, nothing can be concluded in general regarding the convexity of problem (27), as it will depend on properties of the higher-level functional $P_k(x_k(\alpha_k))$.

Note that, as explained in Section 3, when solving (27), one fully runs (till convergence) a parameter choice strategy, with the outcome of selecting a suitable regularization parameter for iteration k (ie, this is in principle a local choice, good for the kth projected Tikhonov problem only): therefore, solving (27) to high precision for every k may be worthless. Moreover, even if only a stopping criterion for the higher-level problem should be set, one needs an additional (and often heuristic) stopping criterion to set k, that is, to guarantee that problem (27) is a good approximation to its full dimensional counterpart , so that one is effectively dealing with two stopping criteria.

Problem (27) can also be regarded as a projected version of the bi-level optimization problem

$$\min_{\alpha \geq 0} P(x(\alpha)) \quad \text{subject to} \quad x(\alpha) = \arg\min_{x \in \mathbb{R}^n} F(x, \alpha) , \tag{29}$$

arising when applying Tikhonov method (4) (lower-level problem) equipped with a given parameter choice rule (higher-level problem) to compute a regularized solution to (1). As for problem (27), nothing can be stated in general about the convexity of problem (29): a couple of instances (one convex and the other nonconvex) will be considered in Sections 4.2 and 4.3 (respectively); see also [74] and the references therein. Again, thanks to the particular form of $F(x, \alpha)$, one can derive a closed-form solution for $x(\alpha)$, and substitute its expression in $P(x(\alpha))$, so that problem (29) is essentially a single-level optimization problem. However, in practice and as explained in Section 1, one can obtain $x(\alpha)$ directly

only when some factorizations of A (such as the SVD) can be computed: this is not the case for large-scale unstructured problems (1). In these situations, one should resort to an iterative linear solver to approximate the solution $x(\alpha)$ of the lower-level problem in (29), while a nonlinear solver is used to compute an approximation to the higher-level problem in (29). Because of this, problem (29) should still be treated as a bi-level optimization problem. In particular, an inner-outer iterative scheme may be naturally established when solving (29). As partially explained in Section 3, this may be the case for *first-regularize-then-project* methods (19), which iteratively solve a lower level problem of the form (4) for every value of the Tikhonov regularization parameter α computed within the iterations of $\min\alpha \geq 0 P(x(\alpha))$. This approach is equivalent to applying the well-known variable projection method [75] for minimizing problems over two sets of dependent variables, that is, in general, one set of unknowns is explicitly eliminated, and the minimization is computed for the resulting variable projection functional that depends only on the remaining variables. In the case of (29), for the unknown variables x and α, $x = x(\alpha)$ can be computed for a given α by iteratively solving a problem of the form (4), so that the minimization is performed over the variable α for the variable projection functional $P(x(\alpha))$. Note that the intermediate values of α so determined are subsequently computed for the full-dimensional problem (4), and two stopping criteria are involved: one for the inner iterations (to be repeatedly applied), and one for the outer iterations.

This section introduces a new efficient class of parameter choice strategies for large-scale problems (4), which can be viewed in the framework of (29), but leverage ideas typical of the hybrid approach to (27). The new strategies simultaneously compute a value for k, α_k and $x_k(\alpha_k)$, thereby computing a good approximation of the solution of the original problem (29).

The core idea behind the new strategies is to "interlace" the iterations needed to solve the lower-level problem and the higher-level problem in (29). These strategies result in only one iteration cycle, bypassing both the *first-regularize-then-project* (19) and *first-project-then-regularize* approaches (20). Namely, each iteration of the new methods consists in performing one step of a Krylov projection method for solving the linear lower-level problem in (29), and one step of a nonlinear (eg, Newton-like) iterative scheme for solving the nonlinear higher-level problem in (29) (as sketched in Algorithm 1).

Algorithm 1. New adaptive bi-level optimization algorithm for problem (29)

1: Choose an initial guess α_1.
2: **for** $k = 1, 2, \ldots$ until a stopping criterion is satisfied **do**
3: Compute the Krylov subspace $\mathcal{K}_k$ and project problem (4).
4: Apply a step of a nonlinear solver to compute α_{k+1} (given $\mathcal{K}_k$ and α_k).
5: **end for**
6: Take $x_k(\alpha_{k+1}) \in \mathcal{K}_k$ as an approximation of the solution of (4).

In this way, the approximation subspace for the solution of (29) is enlarged while a suitable value for the Tikhonov regularization parameter is set. Also, by avoiding nested iteration cycles, only one stopping criterion should be set (this is typically a standard stopping criterion applied to the higher-level problem in (29)). The new strategy, in addition to being conceptually simpler, potentially allows for great computational savings, even when compared with hybrid methods (27). Indeed the latter, for each k, still requires the repeated solution of the lower-level problem (27), which may become expensive when k increases; the following sections report more detailed estimates of the involved computational costs. Although a similar approach was proposed in [56] for the discrepancy principle, the present strategies are new in that they are casted in the framework of bi-level optimization problems (29), and can work in connection with every parameter choice rule of the form (29).

In the following, details of Algorithm 1 will be only developed for projection methods that are built upon the Golub-Kahan bidiagonalization (GKB) algorithm. This is because approximations of the functionals (listed in Table 2) associated to the higher-level problem can be obtained by exploiting the connections between GKB and Gaussian quadrature rules (see Section 4.1). However, the same approach could be used in connection with other projection methods, provided that approximations of the higher-level functional $P(x(\alpha))$ are available, can be computed cheaply at each iteration k, and converge to $P(x(\alpha))$ as k increases. When using GKB, and when the functional $P(x(\alpha))$ in (29) is associated with the discrepancy principle, it can be proved that the regularized solution computed by the new strategy converges to the solution $x(\alpha)$ of (29) (see Section 4.2); when considering other higher-level functionals, no conclusions about convergence can be drawn at this stage (see Section 4.3).

4.1 | Golub-Kahan bidiagonalization (GKB) and its connections to Gaussian Quadrature rules

A pivotal remark that will be exploited in this section is that the symmetric Lanczos algorithm [[22], Chapter 6] and the GKB algorithm are closely related. Indeed, multiplying the second equation in (17) from the left by A, and using the first equation in (17), one obtains

$$AA^TU_k = AV_kB_k = U_k \underbrace{B_kB_k^T}_{=:T_k} + \zeta_{k+1}u_{k+1}e_k^TB_k^T = U_kT_k + \zeta_{k+1}\rho_k u_{k+1}e_k^T \ . \tag{30}$$

Here, the lower bidiagonal matrix B_k defined in (18) can also be regarded as the Cholesky factor of the symmetric positive definite tridiagonal matrix $T_k = B_kB_k^T$ obtained after k iterations of the symmetric Lanczos algorithm applied to AA^T with initial vector b. Moreover, multiplying the first expression in (17) from the left with A^T, and using again the second equation in (17), one obtains

$$\begin{aligned} A^TAV_k &= A^TU_kB_k + \zeta_{k+1}A^Tu_{k+1}e_k^T = A^TU_{k+1}\overline{B}_k = V_{k+1}B_{k+1}^T\overline{B}_k \\ &= V_k \underbrace{\overline{B}_k^T\overline{B}_k}_{=:\hat{T}_k} + \rho_{k+1}\zeta_{k+1}v_{k+1}e_k^T, \end{aligned} \tag{31}$$

so that V_k can be regarded as the matrix generated by performing k steps of the symmetric Lanczos algorithm applied to A^TA, with initial vector A^Tb. After computing the QR-factorization $\overline{B}_k = Q_k\hat{B}_k^T$, where $\hat{B}_k \in \mathbb{R}^{k\times k}$ is lower bidiagonal, one can see that $\hat{B}_k^T$ is the Cholesky factor of the symmetric positive definite tridiagonal matrix $\hat{T}_k = \overline{B}_k^T\overline{B}_k = \hat{B}_k\hat{B}_k^T$.

Now, denote by $C \in \mathbb{R}^{p\times p}$ a generic symmetric positive semidefinite matrix, having spectral decomposition $C = W\Lambda W^T$, where Λ is a diagonal matrix whose diagonal elements are the eigenvalues $0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_p$ of C, and W is the orthonormal matrix whose columns are the normalized eigenvectors of C. The remaining part of this section presents a strategy to compute bounds for general quadratic forms

$$f(\phi, C, u) = u^T\phi(C)u \ , \tag{32}$$

where $u \in \mathbb{R}^p$ is a given vector and ϕ is a given smooth function on the interval $[0, +\infty)$ of the real line. Using standard definitions and derivations, (32) can be expressed as

$$f(\phi, C, u) = u^T\phi(C)u = u^TW\phi(\Lambda)W^Tu = \sum_{i=1}^{p}\phi(\lambda_i)(W^Tu)_i^2 = \int_0^{+\infty}\phi(\lambda)d\omega(\lambda) =: I(\phi) \ . \tag{33}$$

The last equality comes from considering the sum as a Riemann-Stieltjes integral, where the distribution function ω is a non-decreasing step function with jump discontinuities at the eigenvalues λ_i. The chain of equalities (33) makes it natural to consider quadrature rules to approximate the quadratic form in (32). Gaussian quadrature rules will be employed for this purpose, and they will be computed applying the symmetric Lanczos algorithm to C with initial vector u. In Section 4, only quadratic forms of the kind $b^T\phi(AA^T)b$ and $(A^Tb)^T\phi(A^TA)(A^Tb)$ have to be approximated, so that only the symmetric Lanczos algorithm applied to $AA^T \in \mathbb{R}^{m\times m}$ with initial vector $b \in \mathbb{R}^m$, or applied to $A^TA \in \mathbb{R}^{n\times n}$ with initial vector $A^Tb \in \mathbb{R}^n$, have to be considered: this is done implicitly by applying the breakdown-free GKB algorithm to $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ (see assumption (16) and equations (30) and (31)).

Let $T_k = B_kB_k^T \in \mathbb{R}^{k\times k}$ be the symmetric positive definite tridiagonal matrix appearing in (30), produced after performing $k \le \min\{n, m\}$ steps of the Lanczos algorithm applied to the matrix AA^T with initial vector b. Let $\{q_i(\lambda)\}_{i=0}^k$ be the family of orthonormal polynomials with respect to the inner product induced by the measure $\omega(\lambda)$ (associated with AA^T and b), and let $T_k = Y_k\Theta_kY_k^T$ be the spectral decomposition of T_k, where Y_k is the orthonormal matrix whose columns are the normalized eigenvectors of T_k, and Θ_k is the diagonal matrix of eigenvalues $0 < \theta_1 \le \cdots \le \theta_k$. It is well-known that

the k-point Gauss quadrature rule with respect to the measure $\omega(\lambda)$, defined as

$$\mathcal{G}_k(\phi, AA^T, b) := \sum_{j=1}^{k} \phi(\theta_j) \underbrace{||b||^2 (e_1^T (Y_k) e_j)^2}_{=\mu_j} = ||b||^2 e_1^T Y_k \phi(\Theta_k) Y_k^T e_1 = ||b||^2 e_1^T \phi(T_k) e_1 \,, \tag{34}$$

approximates (33) with $C = AA^T$ and $u = b$. More specifically, the eigenvalues of T_k are the zeros of the polynomial $q_k(\lambda)$, as well as the quadrature nodes, and the quadrature weights μ_j are given by the rescaled and squared first components of the eigenvectors of T_k [51]. Analogously, the k-point Gauss-Radau quadrature rule with one assigned node at the origin and with respect to the measure $\omega(\lambda)$, approximating (33) with $C = AA^T$ and $u = b$, can be obtained by suitably modifying the symmetric Lanczos process to compute a symmetric positive semidefinite matrix $\overline{T}_k$ of order k with one prescribed eigenvalue at the origin. This amounts to taking $\overline{T}_k = \overline{B}_{k-1} \overline{B}_{k-1}^T \in \mathbb{R}^{k \times k}$, where $\overline{B}_{k-1}$ is the $(k-1) \times k$ version of the matrix $\overline{B}_k$ in (18) (or, alternatively, is the matrix obtained by selecting the first $(k-1)$ columns of the Cholesky factor B_k of T_k); see [60] for a proof. Eventually, such a quadrature rule reads

$$\mathcal{R}_k(\phi, AA^T, b) := \sum_{j=1}^{k} \phi(\overline{\theta}_j) ||b||^2 (e_1^T (\overline{Y}_k) e_j)^2 = ||b||^2 e_1^T \overline{Y}_k \phi(\overline{\Theta}_k) \overline{Y}_k^T e_1 = ||b||^2 e_1^T \phi(\overline{T}_k) e_1 \,, \tag{35}$$

where $\overline{T}_k = \overline{Y}_k \overline{\Theta}_k \overline{Y}_k^T$ is the spectral decomposition of $\overline{T}_k$, with $\overline{Y}_k$ orthonormal and $\overline{\Theta}_k = \mathrm{diag}(\overline{\theta}_1, \dots, \overline{\theta}_k)$, $0 = \overline{\theta}_1 < \overline{\theta}_2 \leq \dots \leq \overline{\theta}_k$.

Now let $\hat{T}_k = \hat{B}_k \hat{B}_k^T \in \mathbb{R}^{k \times k}$ be the symmetric positive definite tridiagonal matrix appearing in (31), produced after performing $k \leq \min\{n, m\}$ steps of the Lanczos algorithm applied to $A^T A$ with initial vector $A^T b$. Similarly to the previous derivations, the k-point Gauss quadrature rule with respect to the measure $\omega(\lambda)$ (associated with $A^T A$ and $A^T b$), defined as

$$\mathcal{G}_k(\phi, A^T A, A^T b) := \sum_{j=1}^{k} \phi(\hat{\theta}_j) ||A^T b||^2 (e_1^T (\hat{Y}_k) e_j)^2 = ||A^T b||^2 e_1^T \phi(\hat{T}_k) e_1 \,, \tag{36}$$

approximates (33) with $C = A^T A$ and $u = A^T b$. Here, using notations analogous to the previous ones, $\hat{T}_k = \hat{Y}_k \hat{\Theta}_k \hat{Y}_k^T = \hat{Y}_k \mathrm{diag}(\hat{\theta}_1, \dots, \hat{\theta}_k) \hat{Y}_k^T$ is the spectral decomposition of the matrix $\hat{T}_k$. Finally, the matrix $\bar{\hat{T}}_k := \bar{\hat{B}}_{k-1} \bar{\hat{B}}_{k-1}^T \in \mathbb{R}^{k \times k}$, where $\bar{\hat{B}}_{k-1}$ is the matrix constructed by selecting the first $k-1$ columns of the Cholesky factor $\hat{B}_k$ of $\hat{T}_k$, is symmetric positive semidefinite with one prescribed eigenvalue at the origin. Therefore, the k-point Gauss-Radau quadrature rule with respect to the measure $\omega(\lambda)$ (associated with $A^T A$ and $A^T b$), with one node assigned at the origin, can be expressed as

$$\mathcal{R}_k(\phi, A^T A, A^T b) := \sum_{j=1}^{k} \phi(\bar{\hat{\theta}}_j) ||A^T b||^2 (e_1^T (\overline{\hat{Y}}_k) e_j)^2 = ||A^T b||^2 e_1^T \phi(\bar{\hat{T}}_k) e_1 \,, \tag{37}$$

where $\bar{\hat{T}}_k = \overline{\hat{Y}}_k \bar{\hat{\Theta}}_k \overline{\hat{Y}}_k^T = \overline{\hat{Y}}_k \mathrm{diag}(\bar{\hat{\theta}}_1, \dots, \bar{\hat{\theta}}_k) \overline{\hat{Y}}_k^T$ is the spectral decomposition of the matrix $\bar{\hat{T}}_k$.

Assuming that ϕ is a $2k$-times differentiable function, the quadrature errors $\mathcal{E}_{\mathcal{Q}_k}(\phi) = I(\phi) - \mathcal{Q}_k(\phi)$ associated with k-point Gauss and Gauss-Radau quadrature rules (with $\mathcal{Q}_k(\phi) = \mathcal{G}_k(\phi)$ and $\mathcal{Q}_k(\phi) = \mathcal{R}_k(\phi)$, respectively, and where the dependence on the matrices $AA^T, A^T A$, and the vectors $b, A^T b$, has been removed in the interest of generality), are given by

$$\mathcal{E}_{\mathcal{G}_k}(\phi) = \frac{\phi^{(2k)}(\zeta_{\mathcal{G}_k})}{(2k)!} \int_0^{+\infty} \prod_{i=1}^{k} (t - \zeta_i)^2 d\omega(t) \tag{38}$$

and

$$\mathcal{E}_{\mathcal{R}_k}(\phi) = \frac{\phi^{(2k-1)}(\overline{\zeta}_{\mathcal{R}_k})}{(2k-1)!}\int_0^{+\infty} t\prod_{i=2}^{k}(t-\overline{\zeta}_i)^2 d\omega(t)\ , \tag{39}$$

respectively. Here $\zeta_{\mathcal{G}_k}$, $\overline{\zeta}_{\mathcal{R}_k} \in [\lambda_1, \lambda_{\min\{n,m\}}]$. The ζ_i's denote the nodes of a Gauss quadrature rule (so that $\zeta_i = \theta_i$ for (34) and $\zeta_i = \hat{\theta}_i$ for (36)); the $\overline{\zeta}_i$'s denote the nodes of a Gauss-Radau quadrature rule (so that $\overline{\zeta}_i = \overline{\theta}_i$ for (35) and $\overline{\zeta}_i = \bar{\hat{\theta}}_i$ for (37)). As an immediate consequence of formulas (38) and (39), if the derivatives of the function ϕ have constant sign on $[\lambda_1, \lambda_{\min\{n,m\}}]$, then upper or lower bounds for quadratic forms of the kind $b^T\phi(AA^T)b$ and $b^TA\phi(A^TA)A^Tb$ can be obtained by employing Gauss and Gauss-Radau quadrature rules of the form (34)–(37)

4.2 | A rule based on the discrepancy principle and a new modified Newton method

Some preliminary derivations should be performed to see how Tikhonov regularization (4) equipped with the discrepancy principle (21) fits into the framework of problem (29). Consider the quadratic form

$$P(x(\alpha)) = b^T\psi_{\mathrm{DP}}(AA^T,\alpha)b\ ,$$

where

$$\psi_{\mathrm{DP}}(t,\alpha) = \alpha - \alpha\overline{\varepsilon}^2 + 2t\log((\alpha+t)^{-1}) - t^2(\alpha+t)^{-1}\ , \tag{40}$$

is a function defined for $t \geq 0$, $\alpha > 0$ (see also (33)), and $\overline{\varepsilon} = \varepsilon/\|b\| = \tau\|e\|/\|b\|$ in (1) (see also (21) and Table 2). The first and second derivatives with respect to α of the function $\psi_{\mathrm{DP}}(t,\alpha)$ in (40) read

$$\phi_{\mathrm{DP}}(t,\alpha) := \partial_\alpha\psi_{\mathrm{DP}}(t,\alpha) = \alpha^2(t+\alpha)^{-2} - \overline{\varepsilon}^2 \quad \text{and} \quad \partial_\alpha^2\psi_{\mathrm{DP}}(t,\alpha) = 2\alpha t(t+\alpha)^{-3}\ , \tag{41}$$

respectively. Since $\partial_\alpha^2\psi_{\mathrm{DP}}(t,\alpha) \geq 0$ for $\alpha > 0$, then $\partial_\alpha^2 b^T\psi_{\mathrm{DP}}(AA^T,\alpha)b \geq 0$ for $\alpha > 0$ (ie, $b^T\psi_{\mathrm{DP}}(AA^T,\alpha)b$ is convex as a function of α for $\alpha > 0$). Therefore, solving (29) amounts to solving the nonlinear equation (see [72] for complete derivations)

$$0 = b^T\phi_{\mathrm{DP}}(AA^T,\alpha)b = \alpha^2 b^T(AA^T+\alpha I)^{-2}b - \varepsilon^2 = \|b - Ax(\alpha)\|^2 - \varepsilon^2 \tag{42}$$

with respect to α (which is basically (21) specified for Tikhonov method). Since the continuous function $b^T\phi_{\mathrm{DP}}(AA^T,\alpha)b$ is increasing in α, there exists a unique zero α^* in $(0,\infty)$ provided that

$$\lim_{\alpha\to 0} b^T\phi_{\mathrm{DP}}(AA^T,\alpha)b = -\varepsilon^2 < 0 \quad \text{and} \quad \lim_{\alpha\to+\infty} b^T\phi_{\mathrm{DP}}(AA^T,\alpha)b = \|b\|^2 - \varepsilon^2 > 0\ , \tag{43}$$

where the last inequality obviously holds if $\varepsilon^2 < \|b\|^2$ (this is a reasonable bound for the amount of noise in the data, which will be assumed in the following). Equation (42) agrees with the standard discrepancy principle formulation (21) and, in theory, one can easily apply a zero-finder (eg, Newton method) to compute α^*. Since $\phi_{\mathrm{DP}}(t,\alpha)$ is not convex for $\alpha > 0$, Newton method is not guaranteed to globally converge. As proposed in [55], the simple change of variable $\beta = 1/\alpha$ is performed in (42), so that

$$\hat{\phi}_{\mathrm{DP}}(t,\beta) := (\beta t+1)^{-2} - \overline{\varepsilon}^2 \quad \text{and} \quad \hat{f}_{\mathrm{DP}}(\hat{\phi}_{\mathrm{DP}}, AA^T, b, \beta) := b^T\hat{\phi}_{\mathrm{DP}}(AA^T,\beta)b - \varepsilon^2 \tag{44}$$

are decreasing and convex for $\beta > 0$, and a unique zero β^* exists if conditions analogous to (43) are satisfied. Newton method applied to solve the nonlinear equation (with respect to β)

$$0 = \hat{f}_{\mathrm{DP}}(\hat{\phi}_{\mathrm{DP}}, AA^T, b, \beta) = b^T(\beta AA^T + I)^{-2}b - \varepsilon^2 \tag{45}$$

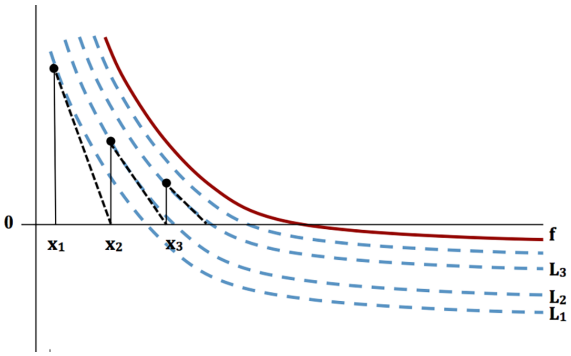


FIGURE 1 Geometrical interpretation of the modified Newton method (47).

globally converges, and can be easily implemented if the SVD of A is available. Since this is not the case in general for large-scale problems (as remarked earlier in this section and in Section 1), an alternative new solution approach for (45) is derived.

The following result proves the convergence of a specific modification of the classical Newton zero finder, which can be used in a general setting whenever dealing with a sequence of functions $\{\mathcal{L}_k\}_{k\geq 1}$ satisfying certain assumptions. This method will be later applied to solve (45).

Theorem 1. *Let $f:(0,+\infty)\to\mathbb{R}$ be a strictly decreasing, convex, differentiable function such that $\lim_{x\to+\infty} f(x) < 0$. Let $\{\mathcal{L}_k\}_{k\geq 1}:(0,+\infty)\to\mathbb{R}$, be a sequence of strictly decreasing, convex, differentiable, increasing lower bounds for f, that is,*

$$\mathcal{L}_k(x) \leq \mathcal{L}_{k+1}(x) \leq f(x) \quad \text{for all} \quad k \geq 1,\ x \in (0,+\infty)\ , \tag{46}$$

such that $\lim_{x\to 0}\mathcal{L}_k(x) > 0$ for all $k\geq 1$, and $\lim_{k\to+\infty}\mathcal{L}_k(x) = f(x)$ for all $x\in(0,+\infty)$. Then, given x_1 such that $\mathcal{L}_1(x_1)\geq 0$, the sequence $\{x_k\}_{k\geq 1}$ obtained from the recursion

$$x_{k+1} = x_k - \frac{\mathcal{L}_k(x_k)}{\mathcal{L}_k'(x_k)} \tag{47}$$

monotonically converges to the root x^* of f from the left.

Proof. The assumptions assure that the functions f and $\mathcal{L}_k$, $k\geq 1$, have exactly one zero in $(0,+\infty)$. Given x_k, $k\geq 1$, define the function

$$t_k(x) = \mathcal{L}_k(x_k) + \mathcal{L}_k'(x_k)(x - x_k)\ ,$$

that is, the tangent line to the graph of $\mathcal{L}_k$ at $(x_k, \mathcal{L}_k(x_k))$. Relation (47) is established by imposing $t_k(x_{k+1}) = 0$ and, together with the convexity of $_k$ and (46), leads to

$$0 = t_k(x_{k+1}) \leq \mathcal{L}_k(x_{k+1}) \leq \mathcal{L}_{k+1}(x_{k+1}) \leq f(x_{k+1}). \tag{48}$$

Replacing k by $k-1$ in the above relation implies $\mathcal{L}_k(x_k)\geq 0$ which, together with $\mathcal{L}_k'(x_k) < 0$ and (47), leads to $x_k \leq x_{k+1}$. Moreover, since f is decreasing and $f(x_{k+1})\geq 0$, $x_{k+1}\leq x^*$. Therefore the sequence $\{x_k\}_{k\geq 1}$ is monotonically increasing and bounded above by x^*. Taking the limit for $k\to\infty$ in (48) implies $\mathcal{L}_{k+1}(x_{k+1})\to f(x_{k+1})$, so that x_{k+1} converges to x^* thanks to the convergence of Newton method. ∎

Remark 1. Given a sequence of functions $\{\mathcal{L}_k\}_{k\geq 1}$, the kth iteration of the modified Newton method (47) consists in performing only one (standard) Newton iteration on the kth function $\mathcal{L}_k$. Figure 1 gives a geometrical illustration of recursion (47).

Remark 2. Theorem 1 still holds if assumption $\mathcal{L}_1(x_1) \geq 0$ is removed and $f, \{\mathcal{L}_k\}_{k\geq 1} : \mathrm{R} \to \mathrm{R}$ (ie, considering functions defined on the whole real line). Indeed, in this setting

$$0 = t_1(x_2) \leq \mathcal{L}_1(x_2),$$

so that the reasoning in the proof of Theorem 1 can be applied starting from x_2.

Turning now to the discrepancy principle (45), since the matrix functional $\hat{f}_{\mathrm{DP}}$ is strictly decreasing, convex, and differentiable with respect to β, and $\partial_t^{(2k)} \hat{\phi}_{\mathrm{DP}}(t, \beta) < 0$ for all $k \geq 1$, $t \geq 0, \beta > 0$, lower bounds for $\hat{f}_{\mathrm{DP}}$ are obtained by applying the Gauss quadrature rule, leading to

$$\mathcal{G}_k(\hat{\phi}_{\mathrm{DP}}, AA^T, b, \beta) = \|b\|^2 e_1^T (\beta B_k B_k^T + I)^{-2} e_1 \ - \varepsilon^2, \quad k \geq 1; \tag{49}$$

see Section 4.1 and equation (38). These bounds are increasing (see [[76], Theorem 2.1]), so that, under assumption (16), and using the shorthand notations $\mathcal{G}_k(\beta)$ and $\hat{f}_{\mathrm{DP}}$ for $\mathcal{G}_k(\hat{\phi}_{\mathrm{DP}}, AA^T, b, \beta)$ and $\hat{f}_{\mathrm{DP}}(\hat{\phi}_{\mathrm{DP}}, AA^T, b, \beta)$, respectively,

$$\mathcal{G}_1(\beta) \leq \mathcal{G}_2(\beta) \leq \cdots \leq \mathcal{G}_{\min\{n,m\}-1}(\beta) \leq \mathcal{G}_{\min\{n,m\}}(\beta) = \hat{f}_{\mathrm{DP}}(\beta) \ .$$

The functions $\mathcal{G}_k(\beta)$, $1 \leq k \leq p$, are strictly decreasing, convex, and differentiable with respect to β, $\lim_{\beta\to 0} \mathcal{G}_k(\beta) = \|b\|^2 - \varepsilon^2 > 0$ (reasonable bound for the amount of noise, see (43)), and $\lim_{\beta\to+\infty} \mathcal{G}_k(\beta) = -\varepsilon^2 < 0$. The same limits hold for $\hat{f}_{\mathrm{DP}}$. The above derivations assure that the assumptions of Theorem 1 are satisfied, so that the following result holds.

Corollary 1. *Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ be as in* (1)*, and let $\hat{\phi}_{\mathrm{DP}}$ be defined as in* (44)*; consider $\hat{f}_{\mathrm{DP}}$ in* (44) *and $\{\ _k\}_k$ in* (49) *as functions of $\beta > 0$. Let $\beta_1 > 0$ be such that $\mathcal{G}_1(\beta_1) \geq 0$. Then the sequence $\{\beta_k\}_{k\geq 1}$ obtained from the recursion*

$$\beta_{k+1} = \beta_k - \frac{\mathcal{G}_k(\beta_k)}{\mathcal{G}_k'(\beta_k)} \tag{50}$$

monotonically converges to the root β^* of (45) *from the left.*

Remark 3. Relation (50) is formally similar to (47). Notably, since $\mathcal{G}_k(\beta) = \hat{f}_{\mathrm{DP}}(\beta)$ for $k \geq \min\{n, m\}$, relation (50) reduces to (standard) Newton method when $k \geq \min\{n, m\}$. However, this never happens in practice, because the bounds $\mathcal{G}_k(\beta)$ are observed to quickly approach $\hat{f}_{\mathrm{DP}}(\beta)$ and the convergence of (standard) Newton method is quadratic; see also Section 5.

The computational cost of the new algorithm depends on the update rule (50) for the value of the regularization parameter for problem (4). At iteration k, $_k$ in (49) is needed to compute α_{k+1} (recall that $\alpha_{k+1} = 1/\beta_{k+1}$): to achieve this, k iterations of the GKB algorithm should be performed (see Section 4.1 and equation (34)) to build the required Krylov subspace $\mathcal{K}_k(AA^T, b)$. Assuming that A is full, the computational cost of this task is dominated by $O(2kmn)$ floating point operations, since two matrix vector products (one with A and one with A^T) are computed at each GKB iteration. Computing $\mathcal{G}_k(\beta_k)$ and $\mathcal{G}_k'(\beta_k)$ in (50) requires the solution of two linear systems with coefficient matrix $(\beta_k B_k B_k^T + I)$ (see also [55]): exploiting the tridiagonal structure of the involved matrices, this amounts to $O(k)$ floating point operations and is therefore negligible. To compute an approximate solution $x_k(\alpha_{k+1})$ for problem (4), one needs to consider the space $\mathcal{K}_k(A^TA, A^Tb)$, and project problem (4) onto it, that is, solve problem (20) (with the Tikhonov regularization parameter equal to $1/\beta_{k+1}$). This can be done inexpensively once the bound $_k$ is computed, since k iterations of the GKB algorithm generate both spaces $\mathcal{K}_k(A^TA, A^Tb)$ and $\mathcal{K}_k(AA^T, b)$ (see Section 4.1), and only the order-k least squares problem in (20) needs to be solved to compute $\hat{y}_k(\beta_{k+1}) := y_k(1/\alpha_{k+1}) \in \mathbb{R}^k$ ($O(k)$ floating point operations) and to then form $\hat{x}_k(\beta_{k+1}) = x_k(1/\alpha_{k+1})$ ($O(kn)$ floating point operations). The cost of these computations is negligible if $k \ll \min\{m, n\}$; moreover this can be performed only once a stopping criterion for the iterations is satisfied. However, $\hat{y}_k(\beta_{k+1}) \in \mathbb{R}^k$ may be needed to devise a suitable stopping criterion, and therefore should be computed at a negligible additional cost at each iteration. Indeed, it should be remarked that the discrepancy functional $\hat{f}_{\mathrm{DP}}$ (44) associated with the approximate solution $\hat{x}_k(\beta_{k+1})$ satisfies

$$\|b\|^2 e_1^T (\beta_{k+1} \overline{B}_k \overline{B}_k^T + I)^{-2} e_1 \ - \varepsilon^2 =: \mathcal{R}_{k+1}(\hat{\phi}_{\mathrm{DP}}, AA^T, b, \beta_{k+1}), \quad k \geq 1 \ ; \tag{51}$$

see (36) (full derivations are provided in [72]). Since $\partial_t^{(2k+1)}\hat{\phi}_{\mathrm{DP}}(t,\beta) > 0$ for all $k \geq 1$, $t \geq 0$, $\beta > 0$, $\mathcal{R}_{k+1}$ is an upper bound for $\hat{f}_{\mathrm{DP}}$ (both considered as functions of β); see (39). Note that (51) is equivalent to (22), that is, the projected version of the discrepancy principle (21) for (direct) Tikhonov regularization.

If, for solving the kth instance of problem (27), one had to adopt a well-established hybrid regularization method based on GKB (for the lower-level problem) and Newton method (for the higher-level problem), $O(2kmn)$ floating points operations should still be needed to compute the partial GKB factorizations (17) (for a full A) and, at each iteration of the (standard) Newton method, two linear systems with tridiagonal coefficient matrix $(\beta_{k+1}\overline{B}_k\overline{B}_k^T + I)$ of dimension $k+1$ should be inverted (see again [55] for more details). If k_N Newton iterations have to be performed, the latter amounts to $O(k_N(k+1))$ floating point operations: although this cost is negligible when compared to the cost associated to GKB, it can become significant (and much higher than the $O(k)$ operations required by the new strategy (50)) when k increases and if many Newton iterations k_N have to be performed.

Summarizing, the kth iteration of the new adaptive strategy to solve problem (29) when $P(x(\alpha))$ is associated with the discrepancy principle consists in applying the new modified Newton zero finder (50) to (45), which involves computing lower bounds $_k$ for $\hat{f}_{\mathrm{DP}}$. The discrepancy functional $\|b - A\hat{x}_k(\beta_{k+1})\|^2 - \varepsilon^2 = \|\overline{B}_k\hat{y}_k(\beta_{k+1}) - \|b\|e_1\|^2 - \varepsilon^2$ for the intermediate approximate solutions $\hat{x}_k(\beta_{k+1}) = x_k(1/\alpha_{k+1})$ of problem (4) lies on upper bounds $\mathcal{R}_{k+1}$ for $\hat{f}_{\mathrm{DP}}$. See also Algorithm 2. An illustration of the behavior of the bounds for the function $\hat{f}_{\mathrm{DP}}$ (44) is given in Section 5 for an image deblurring test problem.

Since the modified Newton method (50) can essentially be regarded as a Newton-like update formula applied to a sequence of iteration-dependent converging functions, standard stopping criteria for Newton method can be adapted to this setting to determine both a value of the regularization parameter β_{k+1} and the dimension of the approximation subspace for $\hat{x}_k(\beta_{k+1})$. Typically, the updates (50) should stop when the space $\mathcal{K}_k(A^TA, A^Tb)$ is large enough to contain a suitable approximation to the solution of (4) and when a value of the regularization parameter suitable for the full dimensional problem (4) has been computed: these requirements are interrelated and, as mentioned in Section 3, they are also desirable for hybrid methods; see [31,61,72].

It is natural to stop the updates (50) as soon as

$$\hat{f}_{\mathrm{DP}}(\beta_{k+1}) = \|b - A\hat{x}(\beta_{k+1})\|^2 - \varepsilon^2 \leq \theta\varepsilon^2 \,, \quad \text{for a given tolerance } \theta > 0 \,. \tag{52}$$

However, computing $\hat{x}(\beta_{k+1})$ would require solving the full-dimensional problem (4) with $\alpha = 1/\beta_{k+1}$, which may be prohibitively expensive for large-scale problems. Therefore, estimates for the function on the left of (52) should be considered. By taking the upper bound (51), one can replace (52) by

$$\mathcal{R}_{k+1}(\hat{\phi}_{\mathrm{DP}}, AA^T, b, \beta_{k+1}) \leq \theta\ \varepsilon^2 \,. \tag{53}$$

Satisfying (53) implies satisfying (52). Alternatively (and using reduced notations), an estimate of $\|b - A\hat{x}(\beta_{k+1})\|^2$ is obtained by averaging its upper (51) and lower (49) bounds evaluated at β_{k+1}, and (52) can be replaced by

$$\frac{1}{2}(\mathcal{R}_{k+1}(\beta_{k+1}) + \mathcal{G}_k(\beta_{k+1})) \leq \theta\ \varepsilon^2 \,. \tag{54}$$

Algorithm 2. Algorithm 1 specified for GKB and the discrepancy principle

1: Choose an initial guess β_1 such that $\mathcal{G}_1(\beta_1) \geq 0$, $\mathcal{G}_k$ as defined in (49).
2: **for** $k = 1, 2, \ldots$ until (53) or (54) are satisfied **do**
3: Apply GKB to update the kth factorizations of the form (17)
4: Compute β_{k+1} by (50)
5: Compute $\hat{y}_k(\beta_{k+1}) = y(1/\alpha_{k+1})$ by (20)
6: **end for**
7: Take $x_k(\alpha_{k+1}) = V_k y_k(\alpha_{k+1}) \in \mathcal{K}_k(A^TA, A^Tb)$ as solution of (27), approximating the solution of (29).

4.3 | Extensions to other parameter choice rules

As already mentioned, the new class of algorithms can be potentially employed to approximate a solution to regularization methods that can be formulated as bilevel optimization problems of the form (29), which includes all the parameter choice rules described in Section 3 used together with Tikhonov regularization (4) (see also Table 2). This section unfolds, in a quite heuristic way, specific derivations for the case of Regińska criterion (26), (28).

As for the discrepancy principle in Section 4.2, the kth iteration of the new approach (starting, in general, from an iteration $k \geq k^*$) consists in performing one minimization step by applying a modified Newton method to the higher level functional $P(x(\alpha))$ in (29), while expanding the approximation subspace for the solution of the lower level functional $F(x, \alpha)$ in (29). Since the evaluation of $P(x(\alpha))$ in (29) at each iteration k is computationally prohibitive, one should employ a sequence of functionals $\mathcal{P}_k(\alpha)$, which have a local minimum converging to a local minimum of $P(x(\alpha))$ but, in general, may not explicitly depend on the current approximate solution $x_k(\alpha)$. In the case of the Regińska criterion, the functionals $\mathcal{P}_k(\alpha)$ are obtained by approximating $P(x(\alpha))$ via Gaussian quadrature rules (see Section 4.1). For this reason, at the kth iteration of the new solver, the Krylov subspaces $\mathcal{K}_k(A^TA, A^Tb)$ and $\mathcal{K}_k(AA^T, b)$ are built, and the new update formula for the Tikhonov regularization parameter reads

$$\alpha_{k+1} = \alpha_k - \frac{\partial_\alpha \mathcal{P}_k(\alpha_k)}{\partial_\alpha^2 \mathcal{P}_k(\alpha_k)}, \quad \text{for} \quad k \geq k^* . \tag{55}$$

Although the above relation is formally similar to (47), where $\mathcal{L}_k = \partial_\alpha \mathcal{P}_k$ and a zero finder is applied to $\partial_\alpha P(x(\alpha)) = 0$, its application may be not as straightforward as in the case considered in Section 4.2.

The functional associated with the Regińska criterion (26) can be expressed in terms of quadratic forms as

$$P(x(\alpha)) = (\alpha^2 b^T \phi_R(AA^T, \alpha)b)^{1/2}((A^Tb)^T \phi_R(A^TA, \alpha)A^Tb)^{\xi/2}, \quad \text{where} \quad \phi_R(t, \alpha) = (\alpha + t)^{-2}, \ \text{and} \ \xi > 0 \ \text{is fixed.} \tag{56}$$

Since $\partial_t^{(2k-1)} \phi_R(t, \alpha) < 0$ for $k \geq 1$, $t \geq 0$, $\alpha > 0$, Gauss-Radau quadrature rules can be used to compute a sequence of increasingly sharper upper bounds for $\alpha^2 b^T \phi_R(AA^T, \alpha)b$ and $(A^Tb)^T \phi_R(A^TA, \alpha)A^Tb$. Namely, taking

$$\begin{aligned} \tilde{\mathcal{R}}_k(\alpha) &:= (\mathcal{R}_{k+1}(\phi_R, AA^T, b, \alpha))^{1/2}(\mathcal{R}_k(\phi_R, A^TA, A^Tb, \alpha))^{\xi/2} \\ &= \|A^Tb\|^\xi \|b\|(e_1^T \phi_R(\overline{B}_k \overline{B}_k^T, \alpha)e_1)^{1/2}(e_1^T \phi_R(\bar{\hat{B}}_{k-1} \bar{\hat{B}}_{k-1}^T, \alpha)e_1)^{\xi/2} \end{aligned} \tag{57}$$

and knowing that both $\mathcal{R}_{k+1}(\phi_R, AA^T, b, \alpha)$ and $\mathcal{R}_{k+1}(\phi_R, A^TA, A^Tb, \alpha)$ decrease with increasing $k \geq 2$, one gets $\tilde{\mathcal{R}}_{k+1}(\alpha) \leq \tilde{\mathcal{R}}_k(\alpha)$ (see Section 4.2 and equations (35), (37)). Similarly, since $\partial_t^{(2k)} \phi_R(t, \alpha) > 0$ for all $k \geq 1$, $t \geq 0$, $\alpha > 0$, Gauss quadrature rules can be used to compute lower bounds for $P(x(\alpha))$ in (56). Namely, taking

$$\tilde{\mathcal{G}}_k(\alpha) := (\mathcal{G}_k(\phi_R, AA^T, b, \alpha))^{1/2}(\mathcal{G}_k(\phi_R, A^TA, A^Tb, \alpha))^{\xi/2} = \|A^Tb\|^\xi \|b\|(e_1^T \phi_R(B_k B_k^T, \alpha)e_1)^{1/2}(e_1^T \phi_R(\hat{B}_k \hat{B}_k^T, \alpha)e_1)^{\xi/2} \tag{58}$$

and knowing that

$$\mathcal{G}_{k+1}(\phi_R, AA^T, b, \alpha) \geq \mathcal{G}_k(\phi_R, AA^T, b, \alpha) \quad \text{and} \quad \mathcal{G}_{k+1}(\phi_R, A^TA, A^Tb, \alpha) \geq \mathcal{G}_k(\phi_R, A^TA, A^Tb, \alpha) ,$$

one gets $\tilde{\mathcal{G}}_{k+1}(\alpha) \geq \tilde{\mathcal{G}}_k(\alpha)$ (see equations (34) and (36)). An illustration of the behavior of the upper (57) and lower (58) bounds for Regińska's functional (56) is given in Section 5 for an image deblurring test problem; this example is quite representative of the typical behavior found in other test problems. Because of these features, the choice $\mathcal{P}_k(\alpha) = \tilde{\mathcal{R}}_k(\alpha)$ seems more convenient and will be considered from now on, while $\tilde{\mathcal{G}}_k(\alpha)$ will be used to devise suitable stopping criteria for the new solver. Moreover, since $\mathcal{P}_k(\alpha)$ is defined for $k \geq 2$, it is natural to select $k^* = 2$ in (55). Note also that both the kth upper (57) and lower (58) bounds for Regińska's functional (56) are different from its projection $R_k(x_k(\alpha_k))$ expressed in (28): indeed, using the notation of Gaussian quadrature rules, $R_k(x_k(\alpha_k)) = (\mathcal{R}_{k+1}(\phi_R, AA^T, b, \alpha))^{1/2}(\mathcal{G}_k(\phi_R, A^TA, A^Tb, \alpha))^{\xi/2}$; see [72] for analogous derivations.

As in Section 4.2, the kth step of the update rule (55) requires that k iterations of the GKB algorithm should be performed, so that, if A is full, this task costs $O(2kmn)$ floating point operations. Updating the regularization parameter α_{k+1} using equation (55), with $\mathcal{P}_k(\alpha_k) = \tilde{\mathcal{R}}_k(\alpha_k)$ in (57), requires the solution of four linear systems with tridiagonal coefficient

matrices (two with $(\overline{B}_k\overline{B}_k^T + \alpha_k I)$ and two with $(\bar{\bar{B}}_{k-1}\bar{\bar{B}}_{k-1}^T + \alpha_k I)$), amounting to $O(k)$ floating point operations. The same reasoning applies to the estimation of the cost for the computation of $\tilde{\mathcal{R}}_k(\alpha_{k+1})$ in (57) and in $\tilde{\mathcal{G}}_k(\alpha_{k+1})$ in (58), which is still $O(k)$ floating point operations: this quantities can be useful to devise a stopping criterion for the new strategy. Computing the approximate solution $x_k(\alpha_{k+1}) = V_k y_k(\alpha_{k+1})$ amounts to $O(kn)$ floating point operations, although this can be done once the stopping criterion is satisfied. If one had to solve the kth instance of problem (27) adopting a well-established hybrid regularization method based on GKB, the computational cost would depend on the optimization method adopted for the higher-level problem: for $k \ll \min\{m, n\}$, the $O(2kmn)$ floating points operations needed to compute the partial GKB factorizations (17) would dominate the computational cost; however, k_M iterations of an optimization routine to minimize $P_k(x_k(\alpha_k))$ based on golden section search would require $k_M + 1$ evaluations of the function (28), amounting to $O((k_M + 1)k)$ floating point operations. Although this cost is negligible when compared to the cost associated to GKB, it can become significant (and much higher than the $O(k)$ operations required by the new strategy (55)) when k and k_M increase.

Similarly to Section 4.2, traditional stopping criteria for (standard) Newton method applied to $\partial_\alpha P(x(\alpha)) = 0$ can be adapted to the modified Newton method (55), in such a way that expensive computation of $x(\alpha_{k+1})$ is avoided. Moreover, when $\{\mathcal{P}_k(\alpha)\}_k$ are approximations of $P(x(\alpha))$ of improving quality (ie, when $\mathcal{P}_k(\alpha)$ becomes closer to $P(x(\alpha))$ as k increases), one can also jointly monitor the convergence of $\mathcal{P}_k(\alpha_{k+1})$ to $P(x(\alpha_{k+1}))$ (ie, the convergence of the approximate functionals to the full-dimensional one at the current approximation of α) together with the convergence of α_{k+1} to a zero of $\partial_\alpha \mathcal{P}_k(\alpha)$. This is done by stopping, for instance, as soon as

$$\frac{|\mathcal{P}_k(\alpha_{k+1}) - P(x(\alpha_{k+1}))|}{|P(x(\alpha_{k+1}))|} + \frac{|\partial_\alpha \mathcal{P}_k(\alpha_{k+1})|}{|\mathcal{P}_k(\alpha_{k+1})|} \approx \frac{|\tilde{\mathcal{R}}_k(\alpha_{k+1}) - \tilde{\mathcal{G}}_k(\alpha_{k+1})|}{|\tilde{\mathcal{R}}_k(\alpha_{k+1}) + \tilde{\mathcal{G}}_k(\alpha_{k+1})|} + \frac{|\partial_\alpha \tilde{\mathcal{R}}_k(\alpha_{k+1})|}{|\tilde{\mathcal{R}}_k(\alpha_{k+1})|} < \theta\,, \quad \text{for a given tolerance } \theta > 0\,. \tag{59}$$

The above approximation is specific for the case of the new modified Newton method applied to the upper bounds $\mathcal{P}_k(\alpha) = \tilde{\mathcal{R}}_k(\alpha)$ for the Regińska functional (57), with $P(x(\alpha_{k+1})) \approx 1/2(\tilde{\mathcal{R}}_k(\alpha_{k+1}) + \tilde{\mathcal{G}}_k(\alpha_{k+1}))$, that is, the average of the upper and lower approximations of $P(x(\alpha))$ in $\alpha = \alpha_{k+1}$.

5 | NUMERICAL EXPERIMENTS

This section investigates the performance of the new class of bi-level optimization algorithms on two large-scale test problems modeling imaging applications. All the experiments are performed running MATLAB R2017a and using some of the functionalities available within the MATLAB toolbox *IR Tools* [77][1]. No so-called *inverse crime* will be committed in that, as detailed in each example below, no perfect agreement between the model and the data in the discretized inverse problem will be assumed. The behavior of different regularized solutions to problem (1) is mainly assessed by evaluating the quality of the reconstructed solutions. If x_{true} is available, the relative restoration error (RRE) (8) is computed, as a function of the Tikhonov regularization parameter α and the dimension k of the Krylov subspaces for the approximation of the solution (to highlight this, the notation $\text{RRE}(\alpha, k)$ will be used).

5.1 | Example 1

This example models an image deblurring test problem, involving the exact image X_{true}, of size 512×512 pixels, appearing in false-color in Figure 2, frame (a); X_{true} represents a galaxy and it is publicly available [2]. The image X_{true} is corrupted by applying a spatially invariant Gaussian blur and Gaussian white noise of level $\|e\|/\|b\|$=0.05; the resulting blurred and noisy image is shown in Figure 2, frame (b). In this setting, given a point-spread function (PSF) analytically defined as a 2D Gaussian function (which describes how a single pixel of the test image is deformed), a blurring process is modeled as a 2D convolution of the PSF and the exact discrete finite image X_{true}. Reflexive boundary conditions, which prescribe pixel values outside of X_{true} for the convolution operation to be well-defined, are assigned. To avoid inverse crime, no perfect agreement between the boundary conditions in the forward model and the operator used to generate the example

[1] MATLAB implementations of our methods, which should be used jointly with *IR Tools*, are available at `https://github.com/silviagazzola`.
[2] It appears among the ones in the NRAO repository, obtained by processing data acquired by a ground-based radio-telescope; see `https://public.nrao.edu/gallery/topic/galaxies/`

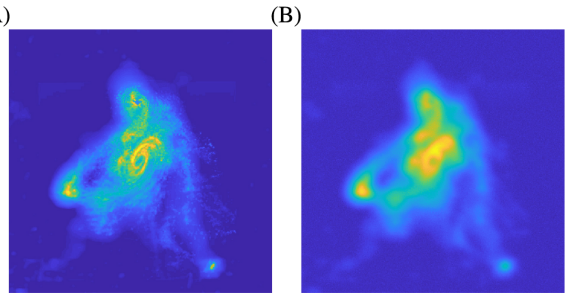


FIGURE 2 Example 1. (a) exact solution (ie, sharp image) X_{true}; (b) noisy and blurred measurements.

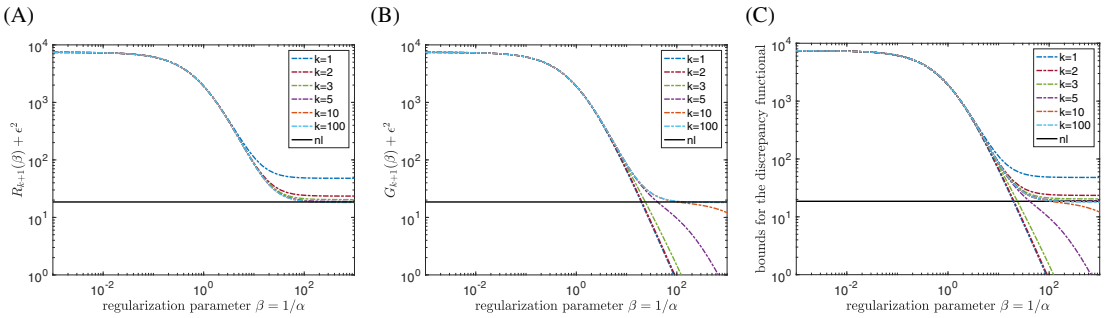


FIGURE 3 Example 1. Bounds for the discrepancy functional $\|b - A\hat{x}(\beta)\|^2$ at different iterations k as a function of the regularization parameter β, displayed in logarithmic scale. (a) upper bounds $\mathcal{R}_{k+1}(\beta) + \varepsilon^2$; (b) lower bounds $\mathcal{G}_k(\beta) + \varepsilon^2$; (c) combination of upper and lower bounds. In all frames the black horizontal line corresponds to the noise magnitude $\varepsilon^2 = \tau^2\|e\|^2$.

is assumed (using the technique described in [[3], §4.6] to simulate a boundary-condition-free real acquisition process). A 2D image restoration problem can be rewritten as a linear system (1), where the square matrix A incorporates the convolution process together with the boundary conditions, the unknown x should approximate the sharp image X_{true} (or, equivalently, the vector x_{true} obtained by stacking the columns of X_{true}), and the vector b is obtained by stacking the columns of the 2D blurred and noisy image; see [3] for more details on image deblurring problems. For this test problem, the coefficient matrix A has order $m = n = 262144$ and it is not explicitly stored; the action of A on any vector is computed efficiently as described in [77].

As extensively explained in Section 4, the new class of algorithms can be applied when upper and lower bounds for the higher-level functional $P(x(\alpha))$ in (29) (or $\partial_\alpha P(x(\alpha))$, that is, its derivative with respect to the parameter α) can be cheaply computed. When $P(x(\alpha))$ is associated with the discrepancy principle (21) or Regińska criterion (26) (with the choice $\xi = 1$), then upper and lower bounds for the relevant functionals are obtained at the kth iteration of the new method by exploiting the links between Gaussian quadrature rules and the GKB algorithm (see Section 4.1). Figures 3 and 4 display such bounds for the discrepancy functional (corresponding to $\partial_\alpha P(x(\alpha))$ in (29)) and the Regińska functional (corresponding to $P(x(\alpha))$ in (29)), respectively, for the considered image deblurring test problem and at different iterations k.

Looking at Figure 3it is immediate to see that both the upper and the lower bounds for the discrepancy functional tend to overlap quite quickly and, in particular, the upper and lower bounds at iteration $k = 100$ essentially coincide for all the considered values of the regularization parameter $\beta = 1/\alpha$ (implying that these bounds already converged to the full dimensional functional $\partial_\alpha P(x(\alpha))$). Looking at Figure 4, it can be observed that both the upper and the lower bounds for the Regińska functional tend to approach one another more slowly than the bounds for the discrepancy functional for every considered value of the regularization parameter; however, they quickly overlap in the rightmost section of the flat region, where the minimum is located. It is important to note that the Regińska functional displays its characteristic flatness around the minimum, which make the minimization problem hard in the full dimensional case (see, eg, [73] for additional examples). This drawback is partially alleviated when considering the upper

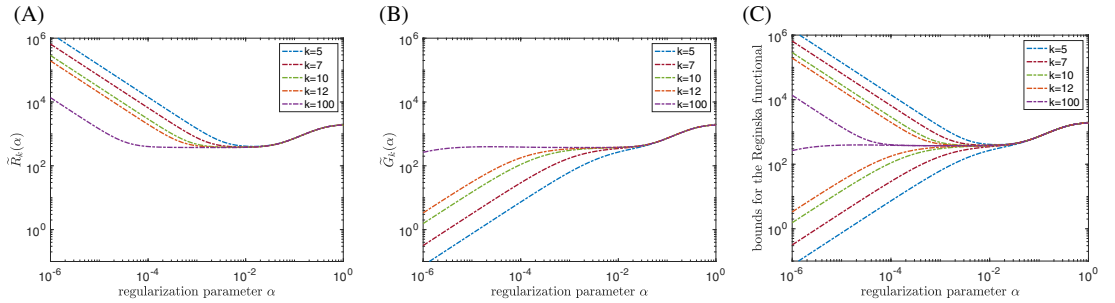


FIGURE 4 Example 1. Bounds for the Regińska functional $\|b - Ax(\alpha)\|/\|x(\alpha)\|$ at different iterations k as a function of the regularization parameter α, displayed in logarithmic scale. (a) upper bounds $\tilde{R}_k(\alpha)$; (b) lower bounds $\tilde{G}_k(\alpha)$; (c) combination of upper and lower bounds.

bounds, which are used in this example to compute the regularization parameter at each iteration via the new adaptive algorithms.

Figure 5 displays: the so-called *discrepancy surface*, that is, a surface whose points lie on $(\alpha, k, \|b - Ax_k(\alpha)\|^2)$; the so-called *Regińska surface*, that is, a surface whose points lie on $(\alpha, k, \|b - Ax_k(\alpha)\|/\|x_k(\alpha)\|)$; the so-called *error surface*, that is, a surface whose points lie on $(\alpha, k, \mathrm{RRE}(\alpha, k))$. Note that, for every fixed k, $\|b - Ax_k(\alpha)\|^2$ corresponds to the upper bound $\mathcal{R}_{k+1}(\alpha) + \varepsilon^2$ for the discrepancy $\|b - Ax(\alpha)\|^2$ (see Section 4.2); however, there is not such correspondence between the projected Regińska functional $\|b - Ax_k(\alpha)\|/\|x_k(\alpha)\|$ and the computed bounds for the Regińska functional displayed in Figure 4 (recall the discussion in Section 4.3). Special markers are used to highlight the points on the different surfaces that are selected by the hybrid and new algorithms used together with the discrepancy principle or Regińska criterion, and the points that deliver a minimal error (ie, an ideally optimal parameter choice rule). Figure 5 gives a good overview on how the discrepancy, the Regińska, and the error functionals evolve across different values of the Tikhonov regularization parameter and across different iterations, and how they relate to each other. For instance, looking at frames (c) and (f), one can see that the optimal value of α (ie, the α minimizing $\mathrm{RRE}(\alpha, k)$) stabilizes for increasing k; correspondingly, looking at frames (a),(b),(d) and (e), one can see that the value of α satisfying the discrepancy principle and the value of α minimizing the Regińska functional stabilize for increasing k. Moreover, as k increases, the parameters selected using the discrepancy principle are essentially optimal (see frames (a) and (d)). It is important to remark that the computation of the discrepancy, the Regińska and the error surfaces merely has illustrative purposes: in practice, the kth iteration of the new algorithm requires evaluating the approximated higher-level functionals for only one value of α once (or twice if a stopping criterion of the form (53) or (54) is implemented, and to compute the approximate solution at the final iteration).

Figure 6 evaluates the performance of the new adaptive bi-level algorithm used in connection with the discrepancy principle (frames (a) and (b)) and Regińska criterion (frames (c) and (d)), by displaying comparisons with other methods. More precisely, the value of the relative restoration error $\mathrm{RRE}(\alpha_{k+1}, k)$ and the value of the regularization parameter α_{k+1} computed by the new method at each iteration k are compared against the values obtained by running (i) a hybrid method with optimal parameter choice at each iteration (so that $\mathrm{RRE}(\alpha, k)$ is minimized with respect to α for each k); (ii) a hybrid method selecting the parameter according to the discrepancy principle at each iteration (in frames (a) and (b)) and the Regińska criterion at each iteration (in frames (c) and (d)); (iii) LSQR used as purely iterative regularization method (ie, without imposing additional Tikhonov regularization within the iterations).

For this test problem, the initial value $\beta_1 = 1/\alpha_1 = 10^{-10}$ is set for the new algorithm used together with the discrepancy principle: this value is small enough to satisfy $\mathcal{G}_1(\beta_1) \geq 0$, so that the assumptions of Corollary 1 hold and convergence to the solution of (29) is guaranteed. On the other hand, the initial value $\alpha_1 = 10^{-3}$ is set for the new algorithm used together with Regińska criterion. Looking at Figure 6 it is evident that the new algorithm used together with either the discrepancy principle or Regińska criterion delivers results that are very similar to the ones obtained when applying their hybrid counterparts (20) (note that the discrepancy principle can be applied in the hybrid framework starting from iteration 7, because equation (22) does not have any zero till that iteration; see, eg, [55]). For this test problem, the performances of both the new algorithm and the hybrid method, used in combination with the discrepancy principle, are very similar to the ones obtained by adopting an optimal regularization parameter at each iteration of a traditional hybrid method. Also, in this example, the combination of a projection method and Tikhonov regularization successfully maintains the

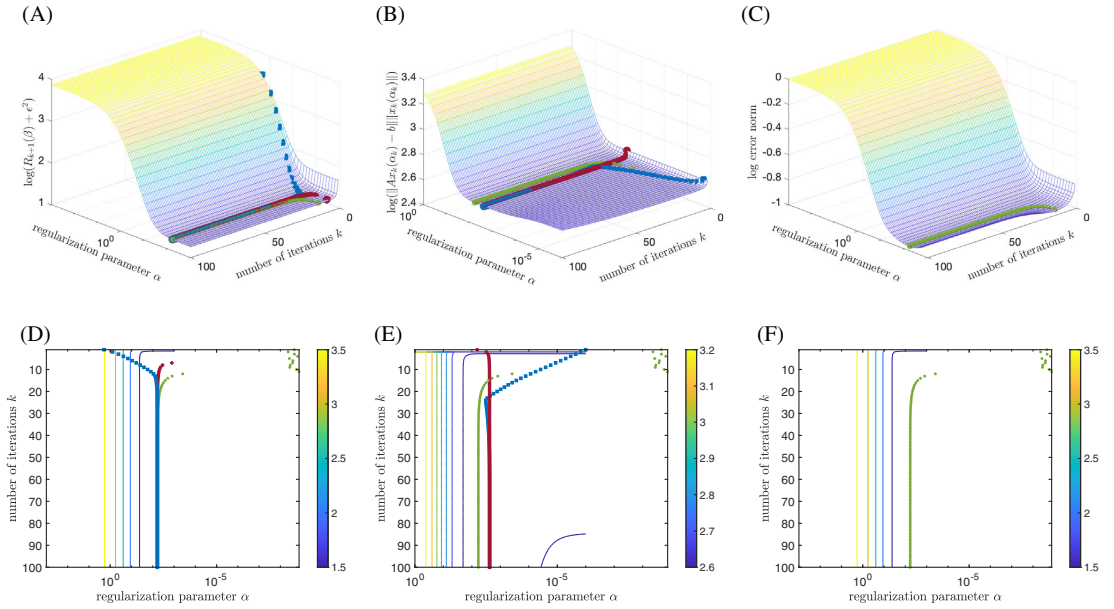


FIGURE 5 Example 1. Two perspectives of different surfaces in logarithmic scale: (a),(d): *discrepancy surface* (ie, the points lying on the discrepancy surface have coordinates $(\alpha, k, \|b - Ax_k(\alpha)\|^2)$) ; (b),(e): *Regińska surface* (ie, the points lying on the Regińska surface have coordinates $(\alpha, k, \|b - Ax_k(\alpha)\| \|x_k(\alpha)\|)$); (c),(f): *error surface* (ie, the points lying on the error surface have coordinates $(\alpha, k, \mathrm{RRE}(\alpha, k))$). In all frames, the blue square markers correspond to the values of the parameters α_k computed at each iteration with the new method (using the discrepancy principle in (a),(d) and Regińska criterion in (b),(e)); the red circular markers correspond to the parameters α_k computed at each hybrid iteration (using the discrepancy principle in (a),(d) and Regińska criterion in (b),(e)), and the green asterisk markers correspond to the values of α_k that deliver a minimal error (optimal) solution (ie, $\min_\alpha \mathrm{RRE}(\alpha, k)$ at each sampled iteration k).

error of the norm at the level of semiconvergence (that affects the purely iterative regularization method LSQR). For both parameter choice strategies, when using the new algorithm, as well as the traditional hybrid algorithms, the values of the regularization parameter (Figure 6 frames (b) and (d)) and the relative restoration error (Figure 6 frames (a) and (c)) stabilize as the iterations proceed. There is perfect agreement between the behavior observed in Figures 5 and 6. Another desirable feature that can be observed in Figure 6 is that the stopping criteria listed in Section 4.2 and 4.3 for the new algorithm succeed in stopping the iterations when a good regularization parameter is computed (ie, at a point that is quite close to the optimal one), and when the relative restoration error is quite low: for this test problem, the threshold $\theta = 10^{-3}$ is set for the discrepancy principle (stopping criteria (53) and (54)), and $\theta = 10^{-1}$ is set for Regińska criterion (stopping criterion (59)). Some stabilization in the values of both the relative restoration errors and the Tikhonov regularization parameters happens when the new algorithm stops according to these stopping criteria; in general, θ can be tuned to find a trade-off between speed and stabilization (to avoid performing additional iterations once some stabilization in the Tikhonov regularization parameters is detected).

Finally, Figure 7 displays some relevant reconstructions computed using different strategies; frame (a) displays the best achievable reconstruction (corresponding to the iteration minimizing the RRE for the optimal parameter choice strategy) and frames (b)-(e) display different reconstructions obtained using the new solvers. Frames (b) and (c) show to the reconstructions at the iterations minimizing the relative reconstruction error for the new solver using the discrepancy principle and the Regińska criterion, respectively. Note that frame (b) also corresponds to the reconstruction for the new algorithm using the discrepancy principle at the iteration given by the stopping criterion (53). At last, frame (d) shows the reconstruction computed by the new solver using the discrepancy principle at the iteration selected by the stopping criteria (54) and frame (e) shows the reconstruction computed by the new solver using Regińska criterion at the iteration selected by the stopping criterion (59). It can be observed that the quality of the reconstruction computed by the new algorithm approaches the optimal achievable quality, and that the stopping criteria successfully pick an early iteration that is close to the best possible iteration.

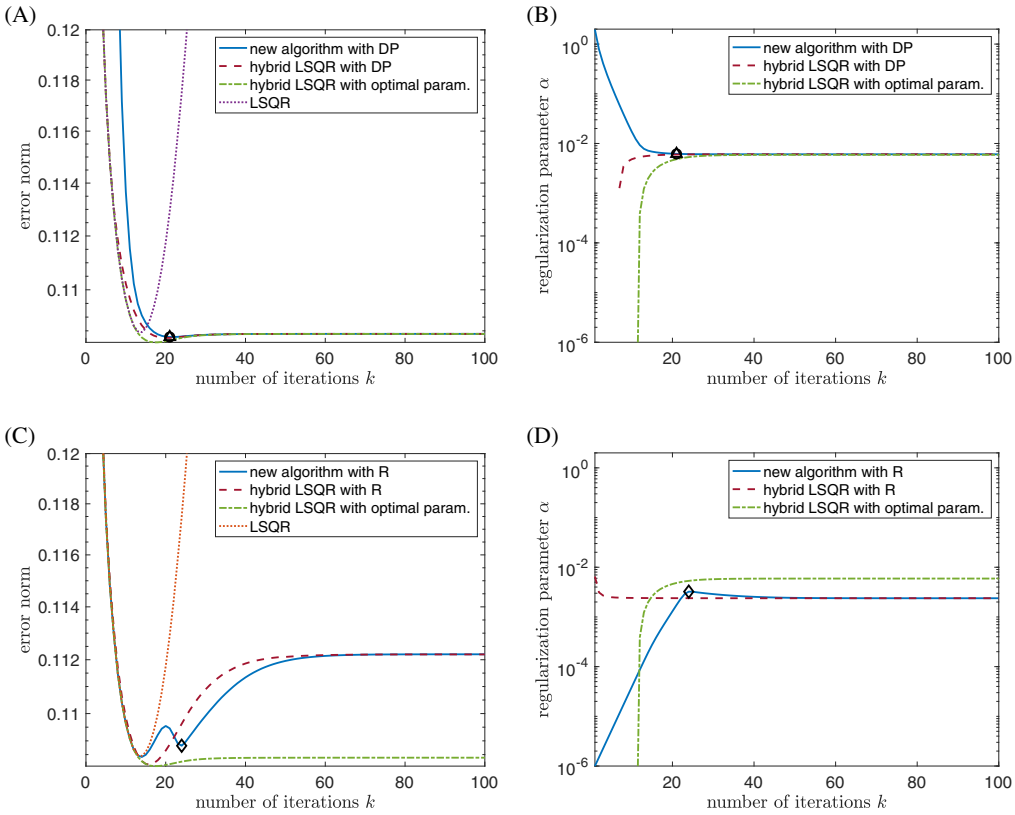


FIGURE 6 Example 1. (a) RRE versus number of iterations k using the discrepancy principle; (b) regularization parameter values versus number of iterations k using the discrepancy principle; (c) RRE versus number of iterations k using Regińska criterion; (d) regularization parameter values versus number of iterations k using Regińska criterion. Circular and triangular markers highlight the iterations satisfying the stopping criterion (53) and (54), respectively, while diamond-shaped markers highlight the iteration satisfying the stopping criteria (59).

5.2 | Example 2

This example uses publicly available data, produced by performing a real tomographic scan of an object (in this case, a carved piece of cheese); see [78]. The data (sometimes referred to as sinogram) consists of measurements of the damping of fan-beam straight X-rays that penetrate the object. More precisely, 180 projections spanning the full 360 degree circle around the object are collected; the inverse problem involves reconstructing an image of the spatially varying attenuation coefficients of a 2D cross-section of the object. This problem leads to a linear system (1), where A is a sparse coefficient matrix of size 178020×65536 (where only approximately 0.8% of its entires are nonzero), x is the vectorialized 2D image (with resolution 256×256 pixels), and b is the vectorialized sinogram. Note that an estimate of the magnitude of the (allegedly Gaussian white) noise e that affects the measurements is not provided.

The following illustrates the performance of the GKB-based Krylov method LSQR (used as a purely iterative regularization method), the GKB-based hybrid methods, and the GKB-based new approach. The maximum number of iterations (ie, the maximum dimension of the Krylov subspace for the approximation of the solution) is 250. Both the discrepancy principle (after a noise estimate has been recovered) and Reginska criterion are used to adaptively set the Tikhonov regularization parameter. Since the solution is expected to be piecewise-constant (as, essentially, only two different attenuation coefficients for the cheese material and for the air in the carvings have to be recovered), a finite difference approximation of the gradient operator is considered as regularization matrix L (whose null space is spanned by constant vectors). More

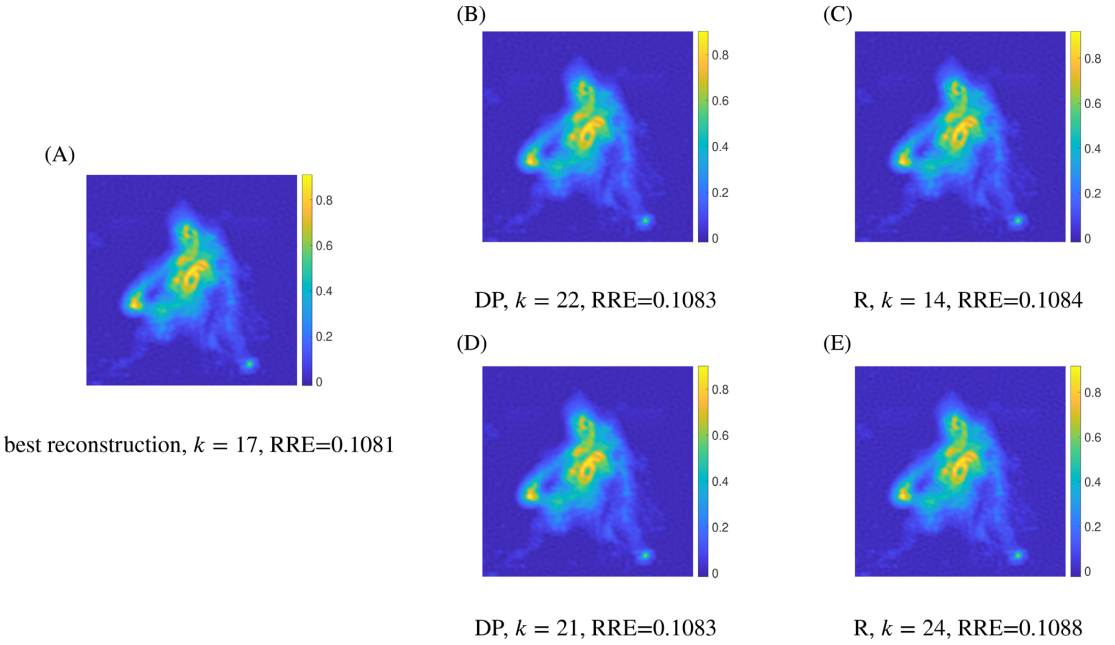


FIGURE 7 Example 1. (a) best achievable reconstruction, corresponding to the iteration minimizing the relative reconstruction error (RRE) for the optimal parameter choice strategy; (b) best reconstruction obtained by the new method used in connection with the discrepancy principle (DP); (c) best reconstruction obtained by the new method used in connection with Regińska criterion (R); (d) reconstruction obtained by the new method used in connection with DP when the stopping criterion (54) is satisfied; (e) reconstruction obtained by the new method used in connection with R when the stopping criterion (59) is satisfied.

precisely, one should take

$$L = \begin{bmatrix} D_1 \otimes I \\ I \otimes D_1 \end{bmatrix} \in \mathbb{R}^{2\sqrt{n}(\sqrt{n}-1)\times n}, \tag{60}$$

where $D_1 \in \mathbb{R}^{(\sqrt{n}-1)\times\sqrt{n}}$ is a finite difference approximation of the one-dimensional first derivative and $I \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$, so that, by exploiting the properties of the Kronecker product $\otimes$, the first and the second blocks of L represent derivatives in the horizontal and vertical directions of a vectorialized 2D image, respectively. As remarked in Section 2, when considering regularizing Krylov methods $L_A^\dagger$ may be formally regarded as right preconditioner that affects the approximation subspace for the solution; in the specific case of the GKB-based methods considered for this example, $x_k \in x_0 + \mathcal{K}_k(L_A^\dagger(L_A^\dagger)^T A^T A, L_A^\dagger(L_A^\dagger)^T A^T b)$, where $x_0 \in \mathcal{N}(L)$. Computationally convenient ways of expressing $L_A^\dagger$, which exploit the structure of (60), are derived in [44]. The following also compares results obtained taking $L = I$ and L as in (60).

The leftmost frame of Figure 8 plots the relative residual norm history (ie, $\|b - Ax_k\|/\|b\|$ versus k) associated to both LSQR (corresponding to regularization in standard form) and preconditioned LSQR (PLSQR, corresponding to regularization in general form, with L as in (60)). One can clearly see that both residuals, despite being non-increasing because of the optimality property (13), somewhat stabilize: this happens very quickly (ie, approximately after 10 iterations) when LSQR is used. Also, the PLSQR residual norm always lays above the LSQR one: this fact reveals that, for this example (and in agreement with the theory hinted in Section 2), PLSQR is actually converging more slowly to the (unregularized) solution of (1). Indeed, looking at the top frames of Figure 9, one can clearly see that LSQR exhibits a quite severe semiconvergence (ie, the reconstruction obtained after 16 iterations is good but, as the iterations advance, a sudden deterioration in the quality of the reconstructions happens and the approximation obtained after 250 iterations is completely meaningless); on the contrary, within the performed 250 iterations, PLSQR seems to somewhat slowly but steadily progress toward

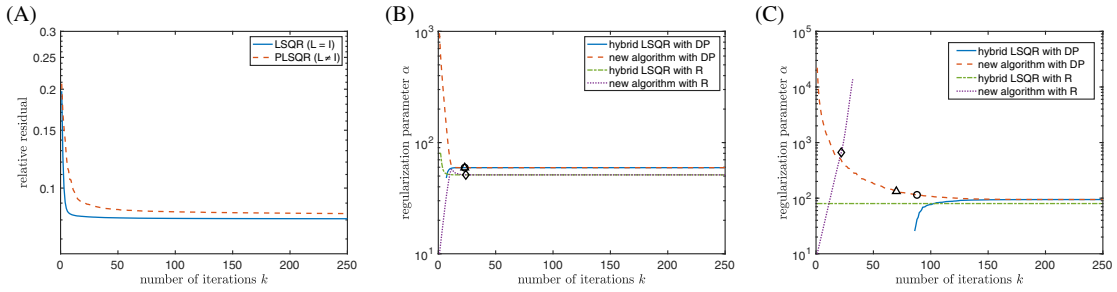


FIGURE 8 Example 2. (a) relative residual norm versus number of iterations k using the purely iterative Krylov solvers LSQR and PLSQR (with preconditioner $LA\dagger$, where L is chosen as in (60)); (b) regularization parameter values versus number of iterations k using the traditional hybrid algorithm and the new algorithm with $L = I$, in connection with the discrepancy principle and Regińska criterion; (c) regularization parameter values versus number of iterations k using the traditional hybrid algorithm and the new algorithm with L as in (60), in connection with the discrepancy principle and Regińska criterion. Circular and triangular markers highlight the iterations satisfying the stopping criteria (53) and (54), respectively, while diamond-shaped markers highlight the stopping criterion (59).

a solution of good quality (ie, the reconstruction obtained after 16 iterations is oversmoothed, but semiconvergence does not happen and the reconstructions are less sensitive to any adopted stopping criterion).

Once LSQR has run, and the norm of the LSQR residual is observed to stabilize, its value can be taken as an estimate of the the magnitude $\|e\|$ of the noise affecting the data; see [79] for more details. Considering now the methods that combine GKB and Tikhonov regularization, the estimate $\|b - Ax_{100}\| \simeq \tau\|e\| = 53.83$ can therefore be used (with a safety factor $\tau = 1.05$) to apply the discrepancy principle; note that, equivalently, the noise level $\|e\|/\|b\|$ for this problem is approximately $8.08 \cdot 10^{-2}$. Alternatively, also Regińska criterion (which does not rely on an estimate of $\|e\|$) can be used, with exponent $\xi = 1$.

The middle frame of Figure 8 displays the values of the Tikhonov regularization parameter selected at each GKB iteration when $L = I$ and when both the new strategy and the classical hybrid approach are employed; both the discrepancy principle and Regińska criterion are used. One can see quite a good agreement between different solvers and regularization parameter choice rules. When the discrepancy principle is used within the hybrid method, equation (22) does not have any zero until iteration 9; the Tikhonov regularization parameter soon stabilizes during the following iterations. When the new strategy is used, a very large Tikhonov regularization parameter is selected trying to satisfy the discrepancy principle during the very early iterations (so that the problem is initially over-regularized), but then, around the 10th iteration, some stabilization happens to the same value of the Tikhonov regularization parameter computed by the hybrid method. Both stopping criteria (53) and (54) with $\theta = 10^{-5}$ are satisfied at the 23rd iteration (but the maximum number of allowed iterations is performed anyway to illustrate the behavior of the new method even after the stoping criteria are satisfied). The (2D versions of the) reconstructions x_k computed by the new method for $k = 23$ and $k = 250$ are displayed in the frames in position (2,1) and (2,2) of Figure 9: the quality of both reconstructions is quite good, meaning that the discrepancy principle (used within the new framework, but also within the classical hybrid framework) selects a suitable regularization parameter, which also allows to overcome semiconvergence. Similarly, when Regińska criterion is used together with the hybrid method and as the iterations proceed, the value of the Tikhonov regularization parameter quickly stabilizes to a value that is slightly lower than the one selected by the discrepancy principle. When the new strategy is used, the parameter selected by Regińska criterion undergoes an initial increase but then, toward the 20th iteration, it decreases and stabilizes on the same value selected by the hybrid method; the stopping criterion (59) with $\theta = 10^{-1}$ is satisfied at the 24th iteration (when, in particular, the value of the parameter is stabilized). The (2D versions of the) reconstructions x_k computed by the new method for $k = 24$ and $k = 250$ are displayed in the frames in position (3,1) and (3,2) of Figure 9: these are almost identical to the ones computed when employing the discrepancy principle (shown in the frames in position (2,1) and (2,2)).

The rightmost frame of Figure 8 displays the values of the Tikhonov regularization parameter selected at each GKB iteration when L is the matrix in (60) and when both the new strategy and the classical hybrid approach are employed; both the discrepancy principle and Regińska criterion are used. Contrarily to the $L = I$ case, one can see several differences between distinct solvers and regularization parameter choice rules. When the discrepancy principle is used within the hybrid method, equation (22) does not have any zero until iteration 86 (which is to be expected because, as displayed

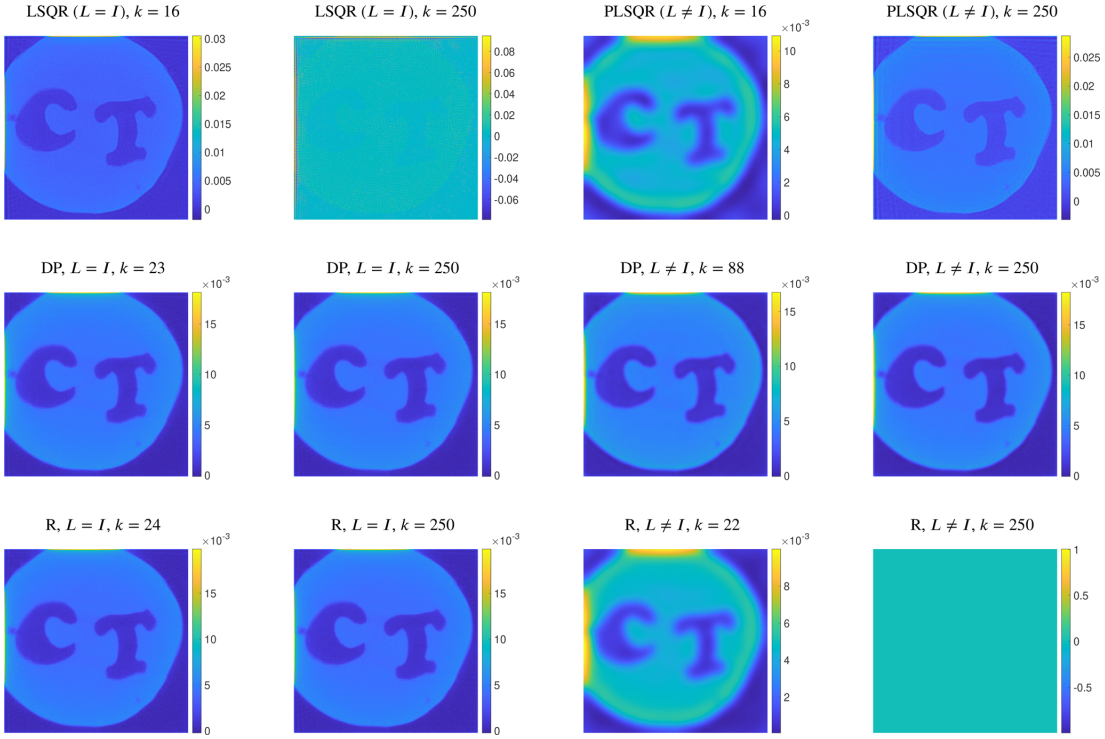


FIGURE 9 Example 2. Reconstructions achieved by the purely iterative LSQR and PLSQR methods (first row) and by the new algorithm used together with the discrepancy principle (DP) (second row) or Regińska criterion (R) (third row). Frames in columns 1 and 2 consider $L = I$, while frames in columns 3 and 4 take L as in (60). Frames in columns 1 and 3 display early iterations or iterations satisfying the stopping criterion ((54) for DP, (59) for R), while frames in columns 2 and 4 display the last iteration.

in the leftmost frame of Figure 8, the PLSQR residual norm decreases much more slowly than the LSQR one); starting from approximately the 200th iteration, the value of the Tikhonov regularization parameter stabilizes to a value much larger than the one selected for the $L = I$ case: this has a desirable effect on the ideally piecewise-constant reconstruction, as the regularization matrix (60) enforces constant reconstructions. When the new strategy is used, after approximately 150 iterations, some stabilization happens to the same value of the Tikhonov regularization parameter computed by the hybrid method. Taking $\theta = 10^{-5}$, the stopping criterion (53) is satisfied at the 70th iteration, while stopping criterion (54) is satisfied at the 88th iteration (also in this case, the maximum number of allowed iterations is performed anyway to illustrate the behavior of the new method even after the stoping criteria are satisfied); one can clearly see that both stopping criteria prescribe to stop before some stabilization in the values of the Tikhonov regularization parameter happens, so that the reconstructions may be slightly over-regularized. The (2D versions of the) reconstructions x_k computed by the new method for $k = 88$ and $k = 250$ are displayed in the frames in position (2,3) and (2,4) of Figure 9: despite the different values $\alpha_{89} = 114.57$ and $\alpha_{251} = 94.52$, visually the two reconstructions are almost indistinguishable. Regińska criterion is not very effective for this instance of this experiment: when used together with the hybrid method, the value of the regularization parameter basically stagnates from the beginning of the iterations; when used together with the new strategy, the value of the regularization parameter steeply increases as the iterations proceed (so that a good stopping criterion is crucial in this situation). This behavior can be explained considering the shape of the functionals $P_k(x_k(\alpha_k))$ (28) to be minimized at each iteration of the hybrid method, and the function $\mathcal{P}_k(\alpha) = \tilde{\mathcal{R}}_k(\alpha)$ to be used in (55). Even if not shown here, $P_k(x_k(\alpha))$ is pathologically flat, making any minimization routine (eg, MATLAB's `fminbnd` that is based on the golden section search) massively reliant on a good estimate of the search interval; indeed, for every iteration k, `fminbnd` performs 500 steps (ie, the default maximum number of steps) and the value of the minimizer barely moves from the starting one. The behavior of the upper bounds $\tilde{\mathcal{R}}_k(\alpha)$ is similar to the one shown in Figure 4 for the image deblurring example, even

if the increase (when α gets large and small) is much steeper, with an an almost constant $\partial_\alpha \mathcal{P}_k(\alpha)$ and an almost zero $\partial_\alpha^2 \mathcal{P}_k(\alpha)$. The stopping criterion (59) with $\theta = 10^{-1}$ for the new method is satisfied at the 22nd iteration, where the selected regularization parameter is much higher than the suitable one selected when the new method based on the discrepancy principle is stopped. For these reasons, one can conclude that, when L is chosen as in (60), Regińska criterion seems to fail for this test problem. However, on the upside, the new method proves to be much more computationally convenient than the hybrid method: adapting the estimates derived at the end of Section 4.3 (exploiting the fact that A is sparse and exploiting the structure of $L_A^\dagger$ when deriving the computational cost of matrix-vector products), one can conclude that performing 250 iterations of the hybrid method requires $O(4.8 \cdot 10^{10})$ flops, while performing 22 iterations of the new method requires $O(4.2 \cdot 10^9)$ flops (ie, approximately 8% of the computations required by the hybrid approach). The (2D versions of the) reconstructions x_k computed by the new method for $k = 22$ (with $\alpha_{23} = 664.27$) and $k = 250$ (with $\alpha_{251} \sim 10^{23}$) are displayed in the frames in position (3,3) and (3,4) of Figure 9: these are both over-smoothed and look quite different from the ones computed when the discrepancy principle is adopted (frames in position (3,1) and (3,2) of the same figure).

6 | CONCLUSIONS AND OUTLOOK

This paper surveyed a variety of iterative regularization methods that are commonly used when obtaining a Tikhonov-regularized solution through direct factorizations would be too computationally demanding, eg, when solving large-scale unstructured linear inverse problems. The main focus of the paper was on regularizing projection methods that are based on Krylov subspace methods, and on the so-called hybrid methods, which combine iterations of a Krylov subspace method and Tikhonov regularization. Hybrid methods are usually preferred to purely iterative regularization methods because, as the iterations progress, if the regularization parameters are properly tuned, the quality of the regularized solution does not deteriorate (and may even improve) at the point of semiconvergence. Well-established but often empirical ways of setting the regularization parameters have been derived in the past decades, and some of them are reviewed in this paper.

This paper also introduced and analyzed a new class of algorithms for the solution of bilevel optimization problems of the form (29) arising when simultaneously computing a Tikhonov-regularized solution and a regularization parameter according to a given rule, and when still considering large-scale unstructured linear inverse problems. By a novel use of Krylov projection methods based on the GKB algorithm, its connections with Gaussian quadrature rules, and a new modified Newton method, the proposed new algorithms "interlace" the iterations performed to apply a given (nonlinear) parameter choice rule and the iterations performed to iteratively solve the (linear) Tikhonov-regularized problem, giving rise to an efficient and principled strategy that delivers results comparable to the ones obtained with traditional hybrid methods.

The introduction of this new class of algorithms carries a number of open questions and possible extensions that may be addressed in future research. For instance, future work can include the natural extension of the new algorithms to work with Krylov projection methods that are based on decompositions other than GKB (eg, the Arnoldi decomposition or flexible Krylov methods; see [80]); also extensions to projection strategies that handle a generic regularization matrix L by computing joint decompositions of the coefficient matrix A and L (see, for instance, [63,81]) may be considered as an alternative to the strategy, employed within this paper, of transforming the Tikhonov problem into standard form and preconditioning the approximation subspace for the solution. Although the algorithmic details of the new approach can be easily adapted to these situations, the theoretical analysis of the resulting strategies needs to be carefully rethought. Also, other parameter choice strategies that can be expressed in the framework of bilevel optimization problems (eg, the UPRE, GCV, and L-curve criteria reported in Section 3) can be considered. Moreover, while a solid converge proof is available for the new algorithm applied in connection with the discrepancy principle, the considered extension to other parameter choice rules is still quite heuristic and more theoretical developments are needed to prove convergence to possible local minima.

Finally, since the new class of algorithms is very general, it may be potentially extended to a variety of bi-level optimization methods that involve the solution of a nonlinear higher-level problem and a linear lower-level problem, even beyond the regularization task. However, as far as regularization is concerned, it should be mentioned that simple constraints such as non-negativity or box constraints, which sometimes have a dramatic effect in enhancing the quality of the reconstructions, cannot be straightforwardly incorporated when adopting the basic Krylov subspace methods discussed in this paper (although some approaches based on flexible Krylov methods are available; see [82]). Similarly, the methods

discussed in this paper cannot be straightforwardly applied to nonlinear ill-posed problems; see, eg, [83,84]. Extending the new class of algorithms to handle these situations can be the focus of future research.

REFERENCES

[1] P. C. Hansen, *Discrete inverse problems: Insight and algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2010.
[2] C. R. Vogel, *Computational methods for inverse problems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002.
[3] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring images: Matrices, spectra, and filtering*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006.
[4] M. Hanke and P. C. Hansen, Regularization methods for large-scale problems, Surveys Math. Indust. **3** (1993), 253–315.
[5] H. W. Engl, M. Hanke, and A. Neubauer, Regularization of inverse problems, in *Mathematics and Its Applications*, Springer, Netherlands, 1996.
[6] J. M. Bardsley and J. G. Nagy, Covariance-preconditioned iterative methods for nonnegatively constrained astronomical imaging, SIAM J. Matrix Anal. Appl. **27** (2006), 1184–1197.
[7] R. A. Renaut, I. Hnětynková, and J. Mead, Regularization parameter estimation for large-scale Tikhonov regularization using a priori information, Comput. Statist. Data Anal. **54** (2010), 3430–3445.
[8] G. H. Golub and C. F. Van Loan, *Matrix computations*, The Johns Hopkins University Press, Baltimore, MD, 1996.
[9] B. Hofman, *regularization for applied inverse and ill-posed problems*, Teubner, Stuttgart, 1986.
[10] P. C. Hansen, The discrete picard condition for discrete ill-posed problems, BIT **30** (1990), 658–672.
[11] J. Chung, G. Easley, and D. P. O'Leary, Windowed spectral regularization of inverse problems, SIAM J. Sci. Comput. **33** (2011), 3175–3200.
[12] L. Dykes and L. Reichel, On the reduction of Tikhonov minimization problems and the construction of regularization matrices, Numer. Algorithms **60** (2012), 683–696.
[13] L. Reichel and Q. Ye, Simple square smoothing regularization operators, Electron. Trans. Numer. Anal. **33** (2009), 63–83.
[14] S. Morigi, L. Reichel, and F. Sgallari, Orthogonal projection regularization operators, Numer. Algorithms **44** (2007), 99–114.
[15] L. Eldén, A weighted pseudoinverse, generalized singular values, and constrained least squares problems, BIT **22** (1982), 487–502.
[16] J. Kamm and J. G. Nagy, Kronecker product and SVD approximations in image restoration, Linear Algebra Appl. **284** (1998), 177–192.
[17] A. Aricó et al., *the anti-reflective transform and regularization by filtering*, in *Lecture Notes in Electrical Engineering*, Vol **80**, Springer-Verlag, Berlin, 2011, 1–21.
[18] S. Gazzola and Landman M. Sabaté, Flexible GMRES for total variation regularization, BIT Numer. Math. **59** (2019), 721–746.
[19] F. Natterer, *The mathematics of computerized tomography*, John Wiley, New York, 1986.
[20] Jia Zhongxiao, Approximation accuracy of the Krylov subspaces for linear discrete ill-posed problems, J. Comput. Appl. Math. **374** (2020), 112786.
[21] P. C. Hansen, *Rank-deficient and discrete ill-posed problems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1998.
[22] Y. Saad, *Iterative methods for sparse linear systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
[23] M. Hanke, Regularization with differential operators: An iterative approach, Numer. Funct. Anal. Optim. **13** (1992), 523–540.
[24] M. Hanke, J. G. Nagy, and R. J. Plemmons, Preconditioned iterative regularization, Numer. Linear Algebra **9** (1992), 141–163.
[25] P. C. Hansen, Regularization, GSVD and truncated GSVD, BIT **29** (1989), 491–504.
[26] D. Calvetti, B. Lewis, and L. Reichel, GMRES-type methods for inconsistent systems, Linear Algebra Appl. **316** (2000), 157–169.
[27] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual method for solving nonsymmetric linear systems, SIAM J. Sci. Comput. **7** (1986), 856–869.
[28] C. C. Paige and M. A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, Association for Computing Machinery. Transactions on Mathematical Software **8** (1982), 43–71.
[29] C. C. Paige and M. A. Saunders, Algorithm 583: LSQR: Sparse linear equations and least squares problems, Association for Computing Machinery. Transactions on Mathematical Software **8** (1982), 195–209.
[30] P. Novati and M. R. Russo, A GCV-based Arnoldi-Tikhonov regularization method, BIT **54** (2014), 501–521.
[31] S. Gazzola, P. Novati, and M. R. Russo, On Krylov projection methods and Tikhonov regularization, Electron. Trans. Numer. Anal. **44** (2015), 83–123.
[32] Y. Huang and Z. Jia, On regularizing effects of MINRES and MR-II for large scale symmetric discrete ill-posed problems, J. Comput. Appl. Math. **320** (2017), 145–163.
[33] S. Gazzola and P. Novati, Inheritance of the discrete Picard condition in Krylov subspace methods, BIT **56** (2016), 893–918.
[34] S. Gazzola, P. Novati, and M. R. Russo, Embedded techniques for choosing the parameter in Tikhonov regularization, Numer. Linear Algebra Appl. **21** (2014), 796–812.
[35] I. Hnětynková, M. Plešinger, and Z. Strakoš, The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data, BIT **49** (2008), 204–220.
[36] I. Hnětynková, M. Kubínová, and M. Plešinger, Noise representation in residuals of CRAIG, LSQR and LSMR regularization, Linear Algebra Appl. **533** (2017), 357–379.
[37] P. Novati, Some properties of the Arnoldi based methods for linear ill-posed problems, SIAM J. Numer. Anal. **55** (2017), 1437–1455.
[38] T. K. Jensen and P. C. Hansen, Iterative regularization with minimum-residual methods, BIT **47** (2007), 103–120.
[39] C. Paige and M. Saunders, Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. **12** (1975), 617–629.
[40] M. Hanke, *Conjugate gradient type methods for ill-posed problems*, CRC Press, Harlow, UK, 1995.

[41] Kilmer M. E., , Stewart G. W. Iterative regularization and MINRES. SIAM J. Matrix Anal. Appl. 2006;**21**:613–628.
[42] D. Calvetti, L. Reichel, and A. Shuibi, Invertible smoothing preconditioners for linear discrete ill-posed problems, Appl. Numer. Math. **54** (2005), 135–149.
[43] M. Donatelli, A. Neuman, and L. Reichel, Square regularization matrices for large linear discrete ill-posed problems, Numer. Linear Algebra Appl. **19** (2012), 896–913.
[44] P. C. Hansen and T. K. Jensen, Smoothing-norm preconditioning for regularizing minimum-residual methods, SIAM J. Matrix Anal. Appl. **29** (2007), 1–14.
[45] P. C. Hansen and T. K. Jensen, Noise propagation in regularizing iterations for image deblurring, Electron. Trans. Numer. Anal. **31** (2009), 669–696.
[46] Å. Björck, E. Grimme, and P. van Dooren, An implicit shift bidiagonalization algorithm for ill-posed systems, BIT **34** (1994), 510–534.
[47] A. Frommer and P. Maass, Fast CG-based methods for Tikhonov-Phillips regularization, SIAM J. Sci. Comput. **20** (1999), 1831–1850.
[48] F. Zama and Piccolomini E. Loli, A descent method for regularization of ill-posed problems, Optim. Methods Softw. **20** (2005), 615–625.
[49] M. E. Kilmer and D. P. O'Leary, Choosing Regularization Parameters in Iterative Methods for Ill-Posed Problems, SIAM J. Matrix Anal. Appl. **22** (2001), 1204–1221.
[50] J. Chung, M. E. Kilmer, and D. P. O'Leary, A framework for regularization via operator approximation, SIAM J. Sci. Comput. **37** (2015), B332–B359.
[51] G. H. Golub and G. Meurant, *Matrices, moments, and quadrature with applications*, Princeton University Press, Princeton, NJ, 2010.
[52] L. Reichel and G. Rodriguez, Old and new parameter choice rules for discrete ill-posed problems, Numer. Algorithms **63** (2013), 65–87.
[53] F. Bauer and M. A. Lukas, Original article: Comparingparameter choice methods for regularization of ill-posed problems, Math. Comput. Simulation **81** (2011), 1795–1841.
[54] V. A. Morozov, On the solution of functional equations by the method of regularization, Soviet Math. Dokl. **7** (1966), 414–417.
[55] L. Reichel and A. Shyshkov, A new zero-finder for Tikhonov regularization, BIT **48** (2008), 627–643.
[56] S. Gazzola and P. Novati, Automatic parameter setting for Arnoldi-Tikhonov methods, J. Comput. Appl. Math. **256** (2014), 180–195.
[57] M. E. Hochstenbach and L. Reichel, An iterative method for Tikhonov regularization with a general linear regularization operator, J. Integral Equations Appl. **22** (2010), 463–480.
[58] R. A. Renaut, S. Vatankhah, and V. E. Ardestani, Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems, SIAM J. Sci. Comput. **39** (2017), B221–B243.
[59] G. Wahba, Practical approximate solutions to linear operator equations when the data are noisy, SIAM J. Numer. Anal. **14** (1977), 651–667.
[60] G. H. Golub and U. Von Matt, Generalized cross-validation for large-scale problems, J. Comput. Graph. Statist. **6** (1997), 1–34.
[61] J. Chung, J. G. Nagy, and D. P. O'Leary, A weighted-GCV method for Lanczos-Hybrid regularization, Electron. Trans. Numer. Anal. **28** (2008), 149–167.
[62] P. C. Hansen and D. P. O'Leary, The use of the L-curve in the regularization of discrete ill-posed problems, SIMA J. Sci. Comput. **14** (1993), 1487–1503.
[63] M. E. Kilmer, P. C. Hansen, and M. I. Español, A projection-based approach to general-form Tikhonov regularization, SIAM J. Sci. Comput. **29** (2007), 315–330.
[64] T. Regińska, A regularization parameter in discrete ill-posed problems, SIAM J. Sci. Comput. **17** (1996), 740–749.
[65] M. F. Hutchinson, A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines, Commun. Stat.: Simul. Comput. **19** (1990), 433–450.
[66] D. A. Girard, A fast 'Monte Carlo cross-validation' procedure for large least squares problems with noisy data, Numer. Math. **56** (1989), 1–23.
[67] C. Fenu, L. Reichel, and G. Rodriguez, GCV for Tikhonov regularization via global Golub-Kahan decomposition, Numer. Linear Algebra Appl. **23** (2016), 467–484.
[68] C. R. Vogel, Non-convergence of the L-curve regularization parameter selection method, Inverse Problems **12** (1996), 535–547.
[69] M. Hanke, Limitations of the L-curve method in ill-posed problems, BIT Numer. Math. **36** (1996), 287–301.
[70] D. Calvetti, L. Reichel, and A. Shuibi, L-curve and curvature bounds for Tikhonov regularization, Numer. Algorithms **35** (2004), 301–314.
[71] D. Calvetti et al., Tikhonov regularization and the L-curve for large discrete ill-posed problems, J. Comput. Appl. Math. **123** (2000), 423–446.
[72] D. Calvetti, G. H. Golub, and L. Reichel, Estimation of the L-curve via Lanczos bidiagonalization, BIT **39** (1999), 603–619.
[73] Bazán Fermín S. Viloche, Fixed-point iterations in determining the Tikhonov regularization parameter, Inverse Problems **24** (2008), 035001.
[74] K. Kunisch and T. Pock, A bilevel optimization approach for parameter learning in variational models, SIAM J. Imaging Sci. **6** (2013), 938–983.
[75] G. Golub and V. Pereyra, Separable nonlinear least squares: the variable projection method and its applications, Inverse Problems **19** (2003), R1.
[76] Lagomasino G. López, L. Reichel, and L. Wunderlich, Matrices, moments, and rational quadrature, Linear Algebra Appl. **429** (2008), 2540–2554.
[77] S. Gazzola, P. C. Hansen, and J. G. Nagy, IR tools: A MATLAB package of iterative regularization methods and large-scale test problems, Numer. Algorithms **81** (2019), 773–811.
[78] Bubba T. A., Juvonen M., Lehtonen J., et al. *Tomographic X-ray data of carved cheese*. arXiv:1705.05732, 2017. https://zenodo.org/record/1254210#.XsT3B8bTWu4.

[79] I. Hnětynková, M. Plešinger, and Z. Strakoš, The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data, BIT **49** (2009), 669–696.
[80] J. Chung and S. Gazzola, Flexible Krylov Methods for ℓ_p Regularization, SIAM J. Sci. Comput. **41** (2019), S149–S171.
[81] M. E. Hochstenbach, L. Reichel, and X. Yu, A Golub-Kahan-type reduction method for matrix pairs, J. Sci. Comput. **65** (2015), 767–789.
[82] S. Gazzola and Y. Wiaux, Fast nonnegative least squares through Flexible Krylov subspaces, SIAM J. Sci. Comput. **39** (2017), A655–A679.
[83] A. B. Bakushinsky, M. Y. Kokurin, and A. Smirnova, *Iterative methods for ill-posed problems: An introduction*, Walter de Gruyter, Berlin, New York, 2011.
[84] B. Kaltenbacher, A. Neubauer, and O. Scherzer, *Iterative regularization methods for nonlinear ill-posed problems*, Walter de Gruyter, Berlin, New York, 2008.

How to cite this article: Gazzola S, Sabaté Landman M. Krylov methods for inverse problems: Surveying classical, and introducing new, algorithmic approaches. *GAMM-Mitteilungen*. 2020;e202000017. https://doi.org/10.1002/gamm.202000017

2.3 Conclusions

This chapter provides a general survey of regularizing projection methods based on standard Krylov subspaces. In particular, it focuses on the combination of such methods with the well-established Tikhonov regularization. As mentioned in Section 1.5.1 of this thesis, the use of Krylov methods in combination with Tikhonov regularization is not new, but [GL20] presents a novel algorithmic framework that automatically sets the regularization parameter at each iteration given a parameter choice rule by incorporating ideas coming from bilevel optimization. When using the discrepancy principle to compute the regularization parameter, [GL20] also includes theoretical guarantees for convergence of the computed solution to the Tikhonov solution with regularization parameter determined by applying the discrepancy principle.

In the next chapter, Krylov subspace methods will be applied to a more general regularization scheme, ℓ_2-ℓ_p regularization, having to resource to modified Krylov subspaces that incorporate variable preconditioning.

Chapter 3

Iteratively Reweighted FGMRES and FLSQR for sparse reconstruction

This chapter presents two new algorithms to compute sparse approximated solutions of large-scale linear discrete ill-posed problems. The new approach is based on the construction of a sequence of quadratic problems approximating ℓ_2-ℓ_p regularization (with additional smoothing to ensure differentiability at the origin) that are partially solved using flexible Tikhonov-Krylov methods. Both algorithms are built upon a solid theoretical justification that guarantees that the sequence of approximate solutions converges to the solution of the smoothed ℓ_2-ℓ_p regularization problem. I present joint work with James Nagy and Silvia Gazzola, which is published in SIAM Journal on Scientific Computing [GNL21].

3.1 Outline of the paper

This paper is concerned with large-scale linear ill-posed inverse problems described in Section 1.2 of this thesis, where the solution x of the linear system (1.2) is known to be sparse. As mentioned in Section 1.4, a popular scheme to compute approximate sparse solutions of (1.2), when considering Gaussian white noise, is to minimize a least squares problem with an ℓ_p-norm as a regularization term for $0 < p \leq 1$.

As explained in Section 1.4, ℓ_2-ℓ_p regularization can be approximated by a sequence of quadratic Tikhonov problems (1.35). Since the regularization matrix W_k is square, diagonal and invertible (this can be assumed through suitable thresholding [HLM$^+$17, WR08a]), we can easily transform problem (1.35) into standard form (1.36). The interpretation of the matrix W_k^{-1} as an iteration-dependent right preconditioner motivates the use of flexible Krylov methods as explained in Section 1.5.3. In this setting, the search space for the solution adaptively incorporates information about the solution, avoiding the nested cycles of iterations. In this paper, we propose two new algorithms, iteratively reweighted FGMRES (IRW-FGMRES) and FLSQR (IRW-FLSQR), that are based on the 'first-regularize-then-project' setting for flexible Krylov methods (i.e., the projected problem (1.41) is considered at iteration k). A theoretical proof of the convergence of the approximate solutions given by both IRW-FGMRES and IRW-FLSQR to the solution of a smoothed version of the ℓ_2-ℓ_p regularization problem is also given. As a note, the smoothing of the ℓ_2-ℓ_p problem ensures

that the functional associated to this regularization scheme is differentiable at the origin, and it is related to the thresholding of the weights (more details are given in [GNL21]). Finally, numerical examples involving image deblurring and computed tomography (described in Section 1.2) are presented to illustrate the performance of IRW-FGMRES and IRW-FLSQR.

Most of the content in the introduction of the paper [GNL21, Section 1] has already been presented in Chapter 1 of this thesis. Therefore, the reader can skip the introduction of the paper without missing important key points. Note that [GNL21, Section 2] contains important remarks about the smoothing of the ℓ_2-ℓ_p functional and the derivations of the quadratic tangent majorants that are not specified in Chapter 1 of this thesis. There are no major notation discrepancies between this thesis and the included paper [GNL21]. It might be worth noting that Chapter 1 of this thesis just considers a general partial factorization of A for either FLSQR or FGMRES, while these are always kept separate in the paper. Moreover, the solution of the projected problem is noted differently, see Table 3.1.

Table 3.1: Notational discrepancies between this thesis and paper [GNL21]

Discrepancies	Thesis	Paper
Solution of the projected flexible Krylov subspace problem	y	$\bar{y}$
Partial decomposition of A for flexible Arnoldi alg.	$AZ_k = U_{k+1}\overline{G}_k$	$AZ_k = V_{k+1}\bar{H}_k$
Partial decomposition of A for flexible Golub-Kahan alg.	$AZ_k = U_{k+1}\overline{G}_k$	$AZ_k = U_{k+1}M_k$

3.2 Published paper

Statement of Authorship

This declaration concerns the article entitled:			
Iteratively Reweighted FGMRES and FLSQR for sparse reconstruction			
Publication status (tick one) **Draft manuscript** ☐ **Submitted** ☐ **In review** ☐ **Accepted** X **Published** ☐			
Publication details (reference)	S. Gazzola, J.G. Nagy, and M. Sabaté Landman. Iteratively reweighted FGMRES and FLSQR for sparse reconstruction. *SIAM J. Sci. Comput.*, (in press).		
Copyright status (tick the appropriate statement) I hold the copyright for this material ☐ Copyright is retained by the publisher, but I have been given permission to replicate the material here X			
Candidate's contribution to the paper (provide details, and also indicate as a percentage)	The theoretical derivations have been mainly produced by the author of the thesis (80%) The author of the thesis has been the main contributor in the presentation of the content (60%) The numerical experiments have been mainly produced by the author of the thesis (80%)		
Statement from Candidate	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature.		
Signed	*Malena Sabaté Landman*	**Date**	10th Feb 2021

Last update: Feb 2019

SIAM J. SCI. COMPUT.
Vol. 0, No. 0, pp. 000–000

ITERATIVELY REWEIGHTED FGMRES AND FLSQR FOR SPARSE RECONSTRUCTION*

SILVIA GAZZOLA†, JAMES G. NAGY‡, AND MALENA SABATÉ LANDMAN†

Abstract. This paper presents two new algorithms to compute sparse solutions of large-scale linear discrete ill-posed problems. The proposed approach consists in constructing a sequence of quadratic problems approximating an ℓ_2-ℓ_1 regularization scheme (with additional smoothing to ensure differentiability at the origin) and partially solving each problem in the sequence using flexible Krylov–Tikhonov methods. These algorithms are built upon a new solid theoretical justification that guarantees that the sequence of approximate solutions to each problem in the sequence converges to the solution of the considered modified version of the ℓ_2-ℓ_1 problem. Compared to other traditional methods, the new algorithms have the advantage of building a single (flexible) approximation (Krylov) subspace that encodes regularization through variable "preconditioning" and that is expanded as soon as a new problem in the sequence is defined. Links between the new solvers and other well-established solvers based on augmenting Krylov subspaces are also established. The performance of these algorithms is shown through a variety of numerical examples modeling image deblurring and computed tomography.

Key words. Krylov methods, flexible Krylov methods, augmented Krylov methods, sparse reconstruction, inverse problems, imaging problems

AMS subject classifications. 65F20, 65F22, 65F30

DOI. 10.1137/20M1333948

1. Introduction.

Large-scale linear ill-posed inverse problems of the form

$$Ax_{true} = b_{true} + e = b, \quad A \in \mathbb{R}^{m\times n}, \tag{1.1}$$

where x_{true} is the desired unknown solution and e is some unknown Gaussian white noise that affects the data b, arise in the discretization of problems stemming from various scientific and engineering applications, such as astronomical and biomedical imaging, or computed tomography in medicine and industry. In particular, we are interested in the case where A is ill-conditioned with ill-determined rank, i.e., the singular values of A decay and cluster at zero without an evident gap between two consecutive ones to indicate numerical rank. In this case, due to the presence of noise in the measured data, the naive solution $A^\dagger b$ of (1.1) (where $A^\dagger$ is the Moore–Penrose pseudoinverse of A) can be very different from the desired solution, $A^\dagger b_{true}$, due to noise amplification; see, e.g., [23]. Therefore, to obtain a meaningful approximation of x_{true}, problem (1.1) should be regularized, i.e., replaced by a closely related problem whose solution is less sensitive to perturbations in the data b (for a more detailed discussion on ill-posed and discrete ill-posed problems and regularization see, e.g., [25]).

*Received by the editors April 27, 2020; accepted for publication (in revised form) November 20, 2020; published electronically DATE.
https://doi.org/10.1137/20M1333948
Funding: The work of the first author was partially funded by EPSRC under grant EP/T001593/1. The work of the second author was partially supported by the U.S. National Science Foundation under grant DMS-1819042. The work of the third author was supported by a scholarship from the EPSRC Centre for Doctoral Training in Statistical Applied Mathematics at Bath (SAMBa) under project EP/L015684/1.
†Department of Mathematical Sciences, University of Bath, Bath BA2 7AY, UK (S.Gazzola@bath.ac.uk, https://people.bath.ac.uk/sg968/, m.sabate.landman@bath.ac.uk, https://people.bath.ac.uk/msl39/).
‡Department of Mathematics, Emory University, Atlanta, GA 30322 USA (jnagy@emory.edu).

One of the best-known approaches for regularizing linear ill-posed problems is Tikhonov regularization, which, in its general formulation, computes a regularized approximation to the solution of (1.1) by solving the following minimization problem:

$$x_{\lambda,L} = \min_x \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2 \,. \tag{1.2}$$

Here, the regularization parameter $\lambda > 0$ balances the effect of the fit-to-data term $\|Ax - b\|_2^2$ and the regularization term $\|Lx\|_2^2$. The regularization matrix $L \in \mathbb{R}^{q \times n}$ has the effect of enhancing certain properties on the solution and it is usually chosen to be the identity (in this case, problem (1.2) is said to be in standard form) or a rescaled finite differences approximation of a derivative operator (to enforce smoother solutions); if the null space of A and the null space of L intersect trivially, the general-form Tikhonov solution $x_{\lambda,L}$ is unique.

For large-scale problems, where A does not have an exploitable structure nor is even explicitly stored (i.e., may be defined as a function that efficiently computes the actions of A and, possibly, A^T on vectors), the only way to solve problem (1.1) is to apply an iterative method to obtain a sequence of approximated solutions $\{x_k\}_{k \geq 1}$. In fact, many well-known general iterative solvers, e.g., Landweber and Kaczmarz methods, and many Krylov subspace methods leverage the so-called semiconvergence phenomenon and lead to a regularized solution if the iterations are stopped sufficiently early, with the number of iterations playing the role of a discrete regularization parameter (see [25, Chapter 6] for a more accurate description). This paper will only consider the GMRES and LSQR iterative methods and variations thereof: these are Krylov methods that compute a regularized solution by expanding an approximation subspace for the solution and solving a projected least squares problem at each iteration. Note that LSQR is mathematically equivalent to CGLS.

When regularization relies on semiconvergence only, a bad stopping criterion can lead to a big error in the approximated solution. Moreover, semiconvergence may happen before the relevant basis vectors for the solution are incorporated in the Krylov approximation subspace for the solution; see [25, Chapter 6] and [28] for more details. These issues can be mitigated by applying further regularization within the iterations, e.g., by using schemes that combine an iterative Krylov solver and Tikhonov regularization, as detailed below. Consider, for simplicity, $L = I$ in (1.2), i.e., Tikhonov regularization in standard form. Projecting (1.2) into a kth dimensional Krylov subspace spanned by the columns of the matrix V_k leads to

$$x_k = V_k y_k, \quad y_k = \arg\min_y \|AV_k y - b\|_2^2 + \lambda \|V_k y\|_2^2, \tag{1.3}$$

which is sometimes referred to as the "first-regularize-then-project" approach [25, Chapter 6]. Alternatively, a "first-project-then-regularize" approach can also be used, which involves projecting the original linear system (1.1) and then applying standard Tikhonov regularization, leading to

$$x_k = V_k y_k, \quad y_k = \arg\min_y \|AV_k y - b\|_2^2 + \lambda \|y\|_2^2. \tag{1.4}$$

For fixed λ, and assuming the columns of V_k to be orthonormal, expressions (1.3) and (1.4) are equivalent and both schemes are interchangeable. Methods employing the latter approach are also known as hybrid methods [11, 37] and they have recently attracted a lot of attention in the case of large-scale problems where the regularization parameter λ is not known a priori; see [10, 19, 21, 30]. Indeed, hybrid methods

allow for a very efficient (local) choice of the parameter $\lambda = \lambda_k$ at each iteration $k \ll \min\{m, n\}$; moreover, when k increases, λ_k seems to stabilize around a value that is suitable for the full-dimensional problem (1.2).

Tikhonov regularization as defined in (1.2) is rather restrictive, and more general regularization strategies can yield to better approximations of the solution of (1.1). In particular, this paper focuses on regularized problems of the form

$$\min_x \|Ax - b\|_2^2 + \lambda \|x\|_p^p \,, \tag{1.5}$$

where, for $0 < p \leq 1$, the ℓ_p-norm regularization term enforces sparsity in the solution. Although sparse vectors have a small ℓ_0 "norm," considering an ℓ_0 regularization term yields to an NP hard optimization problem (1.5); see [16]. Therefore, it is common to approximate the ℓ_0 regularization term by an ℓ_p term with $0 < p \leq 1$, noting that for $0 < p < 1$ problem (1.5) is nonconvex, and for $p = 1$ problem (1.5) approximates the desired ℓ_0-norm via convex relaxation but is nondifferentiable at the origin; see, e.g., [27, 31, 32]. Note that if sparsity of the solution is assumed in a different domain (e.g., wavelets or discrete cosine transform) a sparsity transform can be incorporated in the regularization term. The values $0 < p \leq 2$ will be considered in this paper; when $p = 2$, problem (1.5) reduces to Tikhonov regularization in standard form. The ℓ_2-ℓ_p regularization problem (1.5) can be solved by a variety of optimization methods [4, 22, 33, 46] or by employing iterative schemes that approximate the regularization term in (1.5) by a sequence of weighted ℓ_2 terms [39]. Methods of the second kind come equipped with (local) convergence proofs for most values of $p > 0$ but usually rely on inner-outer schemes so they can become very expensive computationally; see, e.g., [5, Chapter 4].

More recently, solvers for the ℓ_2-ℓ_p regularization problem that avoid nested loops of iterations by combining reweighting techniques and modified Krylov methods have gained popularity. Namely, generalized Krylov subspaces are considered in [31, 27, 6], and hybrid solvers based on the flexible Arnoldi and the flexible Golub–Kahan decompositions are considered in [9, 18, 20].

In this paper, we propose two new iterative Krylov–Tikhonov methods that use the flexible Arnoldi and the flexible Golub–Kahan decomposition, respectively, to solve the ℓ_2-ℓ_p regularization problem (1.5) by building a single approximation subspace through the iterations. Both algorithms are essentially different from the strategies already available in the literature. On the one hand, differently from [31, 27, 6], the approach proposed in this paper is based on flexible Krylov subspaces. On the other hand, differently from the "first-project-then-regularize" scheme corresponding to hybrid methods implicitly adopted in [9, 18], the approach proposed in this paper exploits a "first-regularize-then-project" scheme. In fact, another contribution of this paper is to show that regularizing and projecting are not interchangeable anymore in the flexible Krylov subspace setting, and properties derived from using the "first-regularize-then-project" approach are used to provide theoretical justification of convergence for the newly proposed algorithms. An original interpretation of the new algorithms in the general framework of augmented and recycled Krylov subspaces is also given. It should be stressed that both new algorithms are inherently "matrix-free" (i.e., they only require the action of A on vectors, and additionally the action of A^T if the flexible Golub–Kahan decomposition is considered) and allow for an iteration-dependent choice of the regularization parameter.

The paper is organized as follows. In section 2 background material on ℓ_2-ℓ_p regularization is reviewed. In particular, section 2 explains how to approximate the ℓ_p

regularization term in (1.5) using an iteratively reweighted scheme, and how the transformation of the resulting problem into standard form leads to iteration-dependent right preconditioning for a Tikhonov problem of the form (1.2). In section 3 two new algorithms for sparse reconstruction (called IRW-FGMRES and IRW-FLSQR) are introduced, along with a solid theoretical proof of convergence and links with augmented Krylov subspace methods. Finally, numerical results are presented in section 4, and general conclusions are given in section 5.

2. Background on ℓ_2-ℓ_p regularization. Iteratively reweighted schemes for the ℓ_2-ℓ_p regularization problem intrinsically rely on the interpretation of problem (1.5) as a nonlinear weighted least squares problem of the form

$$\min_x \|Ax-b\|_2^2 + \lambda \|x\|_p^p = \min_x \|Ax-b\|_2^2 + \lambda \|W^{(p)}(x)x\|_2^2, \tag{2.1}$$

where the diagonal weighting $W^{(p)}(x)$ is defined as

$$W^{(p)}(x) = \operatorname{diag}\left(\left(|[x]_i|^{\frac{p-2}{2}} \right)_{i=1,\dots,n} \right), \tag{2.2}$$

and $[x]_i$ denotes the ith component of the vector x. Note that when $0 < p < 2$, division by zero might occur if $[x]_i = 0$ for any $i \in \{1,\dots,n\}$ and, in fact, this is a far from unlikely situation in the case of sparse solutions. For this reason, in this paper, instead of (2.2), the closely related weights

$$\widetilde{W}^{(p,\tau)}(x) = \operatorname{diag}\left(\left(\left([x]_i^2 + \tau^2 \right)^{\frac{p-2}{4}} \right)_{i=1,\dots,n} \right), \tag{2.3}$$

where τ is a fixed parameter chosen ahead of the iterations, are considered, and problem (2.1) is replaced by

$$\min_x \underbrace{\|Ax-b\|_2^2 + \lambda \|\widetilde{W}^{(p,\tau)}(x)x\|_2^2}_{T^{(p,\tau)}(x)}, \tag{2.4}$$

where $\tau \neq 0$ also ensures that $T^{(p,\tau)}(x)$ is differentiable at the origin for $p > 0$. Note that (2.4) should be considered a smooth version of problem (2.1) and, formally, problem (2.1) can be recovered from problem (2.4) setting $\tau = 0$.

A well-established framework to solve problem (2.4) is the local approximation of $T^{(p,\tau)}$ by a sequence of quadratic functionals $T_k(x)$ that give rise to a sequence of quadratic problems of the form

$$x_{k,\star} = \arg\min_x \underbrace{\|Ax-b\|_2^2 + \lambda \|W_k x\|_2^2 + c_k}_{T_k(x)}, \tag{2.5}$$

where $W_k = \widetilde{W}^{(p,\tau)}(x_{k-1,\star})$. Here, c_k (a constant term for the kth problem in the sequence with respect to x) and λ (which has absorbed other possible multiplicative constants with respect to (2.4)) are chosen so that $T_k(x)$ in (2.5) corresponds to a quadratic tangent majorant of $T^{(p,\tau)}(x)$ in (2.4) at $x = x_{k-1,\star}$. By definition, this implies that $T_k(x) \geq T^{(p,\tau)}(x)$ for all $x \in \mathbb{R}^n$, $T_k(x_{k-1,\star}) = T^{(p,\tau)}(x_{k-1,\star})$, and $\nabla T_k(x_{k-1,\star}) = \nabla T^{(p,\tau)}(x_{k-1,\star})$; see also [27, 39]. Since p and τ are chosen ahead of the iterations, they are omitted from the notation for the weighting matrix W_k.

The vector $x_{k,\star}$ formally denotes the solution of (2.5). For moderate-scale problems, or for large-scale problems where A has some exploitable structure, $x_{k,\star}$ may be

obtained by applying a direct solver to (2.5). For large-scale unstructured problems, only iterative solvers can be used in different fashions to approximate the solution of (2.5), naturally leading to an inner-outer iteration scheme for the sequence of problems (2.4). This is the case considered in the present paper, so that $x_{k,\star}$ corresponds to the approximate solution $x_{k,l}$ of the kth problem of the form (2.5) (or "at the kth outer iteration") at the lth iteration of the inner cycle of iterations. Iteratively reweighted least squares (IRLS) and iteratively reweighted norm (IRN) methods based on an inner-outer iteration scheme are very popular [12, 39] and have been used in combination with different inner solvers, such as steepest descent and CGLS. Typically $x_{k,\star} = x_{k,l}$ is obtained when a stopping criterion is satisfied for problem (2.5) to indicate convergence of the approximate solution; alternatively, problem (2.5) can be partially solved and $x_{k,\star} = x_{k,l}$ denotes the latest available approximation of x. In any case, $T_k(x)$ in (2.5) is a quadratic tangent majorant of $T^{(p,\tau)}(x)$ in (2.4) at $x = x_{k-1,\star}$, and IRLS or IRN approaches are particular instances of majorization-minimization schemes: for fixed λ, it is known that solving a sequence of problems of the form (2.5) produces a sequence of approximate solutions that converge to the minimizer of problem (2.4); see, e.g., [12]. Fully solving each problem (2.5) can result in a computationally demanding scheme.

For W_k square and invertible (note that this can be assumed for any fixed $p > 0$ when the weights are defined as in (2.3) with $\tau > 0$), problem (2.5) can be easily and conveniently transformed into standard form as follows:

$$\bar{x}_{k,\star} = \arg\min_{\bar{x}} \|AW_k^{-1}\bar{x} - b\|_2^2 + \lambda\|\bar{x}\|_2^2 \quad \text{so that} \quad x_{k,\star} = W_k^{-1}\bar{x}_{k,\star}. \tag{2.6}$$

The interpretation of the matrix W_k^{-1} as a right preconditioner for problem (2.5) can be exploited under the framework of prior-conditioning [7]. The simplest way to use formulation (2.6) in combination with Krylov methods is to rely on an inner-outer scheme (e.g., with an inner loop of (hybrid) GMRES or LSQR iterations [9, 18]) so that, at each outer iteration, a new Krylov subspace is built. Let $V_{k,l} \in \mathbb{R}^{n\times l}$ be the matrix whose columns, at the lth inner iteration of the kth outer cycle, span a Krylov subspace $\mathcal{K}_{k,l}$ of dimension l. Then, problem (2.6) can be projected and solved in $\mathcal{K}_{k,l}$ by computing

$$\bar{y}_{k,l} = \arg\min_{\bar{y}} \left\| A\overbrace{W_k^{-1}\underbrace{V_{k,l}\bar{y}}_{\bar{x}}}^{x} - b\right\|_2^2 + \lambda\| \underbrace{V_{k,l}\bar{y}}_{\bar{x}} \|_2^2 , \tag{2.7}$$

so that $\bar{x}_{k,l} = V_{k,l}\,\bar{y}_{k,l}$ and $x_{k,l} = W_k^{-1}\,\bar{x}_{k,l} = W_k^{-1}\,V_{k,l}\,\bar{y}_{k,l}$. Note that since $V_{k,l}$ has orthonormal columns, solving (2.7) is equivalent to solving

$$\bar{y}_{k,l} = \arg\min_{\bar{y}} \left\| A\underbrace{\overbrace{W_k^{-1}V_{k,l}}^{x}\,\bar{y}}_{Z_{k,l}} - b\right\|_2^2 + \lambda\|\bar{y}\|_2^2, \tag{2.8}$$

which is consistent with the idea of "first-regularize-then-project" being equivalent to "first-project-then-regularize" for hybrid solvers (cf. [25, Chapter 6]). An alternative interpretation of this scheme is that, at the lth inner iteration of the kth outer cycle, an approximate solution to the original problem is sought in the preconditioned space

$\mathcal{R}(Z_{k,l}) = \mathcal{R}(W_k^{-1}V_{k,l})$, where $\mathcal{R}(\cdot)$ denotes the range of a matrix. Note that, when applying preconditioned GMRES,

$$\mathcal{R}(Z_{k,l}) = W_k^{-1}\mathcal{K}_l\left(AW_k^{-1}, b\right) = \text{span}\left\{W_k^{-1}b, W_k^{-1}\left(AW_k^{-1}\right)b, \ldots, W_k^{-1}\left(AW_k^{-1}\right)^{l-1}b\right\}, \tag{2.9}$$

while, when applying preconditioned LSQR,

$$\mathcal{R}(Z_{k,l}) = W_k^{-1}\mathcal{K}_l\left(W_k^{-1}A^TAW_k^{-1}, W_k^{-1}A^Tb\right) = \text{span}\left\{\left(W_k^{-1}\right)^2A^Tb, , \ldots, \left(\left(W_k^{-1}\right)^2A^TA\right)^{l-1}\left(W_k^{-1}\right)^2A^Tb\right\}. \tag{2.10}$$

With respect to preconditioned GMRES, preconditioned LSQR naturally applies the inverse of the weight matrix W_k twice for every new direction included in the search space and, hence, twice at each iteration.

It should be stressed that for both (2.7) and (2.8) to be equivalent to (2.6), the regularization term in (2.7) has to be $\|V_{k,l}\bar{y}\|_2^2$, where $V_{k,l}\,\bar{y} = \bar{x}$ in (2.6), and not $\|Z_{k,l}\bar{y}\|_2^2$. Using $\|Z_{k,l}\bar{y}\|_2^2$ as a regularization term would in fact be equivalent to solving a different problem, namely, Tikhonov problem (1.2) with the identity as a regularization matrix (i.e., in standard form), in the preconditioned Krylov subspace $\mathcal{R}(Z_{k,l})$. It is important to note that $\mathcal{R}(Z_{k,l})$ incorporates regularization through preconditioning.

Flexible Krylov methods provide a natural framework to efficiently avoid nested loops of iterations by regarding the inverse of the regularization matrix (stemming from an iteratively reweighted regularization term) as iteration-dependent right preconditioning in (2.6). In this setting, at the kth iteration, the weights W_k are updated using the most recent approximation of the solution, i.e., the one at the $(k-1)$th iteration of the flexible solver, and incorporated in the construction of the flexible Krylov space in the form of the adaptive preconditioner W_k^{-1}. Flexible Krylov subspaces based on either the flexible Arnoldi or the flexible Golub–Kahan decompositions are summarized below.

Flexible Arnoldi decomposition. The flexible Arnoldi decomposition of $A \in \mathbb{R}^{n\times n}$ was first introduced in [40], and it is commonly employed in different settings to incorporate adaptive or increasingly improved preconditioners into the solution subspace; see [42, Chapter 9] and [43, 44]. Given A (square), b and right iteration-dependent preconditioning matrices W_k^{-1}, the partial factorization

$$AZ_k = V_{k+1}\bar{H}_k \tag{2.11}$$

is updated at iteration k (for $k \le n$), where $\bar{H}_k \in \mathbb{R}^{(k+1)\times k}$ is upper Hessenberg, V_{k+1} has orthonormal columns with $v_1 = b/\|b\|_2$, and $Z_k = [W_1^{-1}v_1, \ldots, W_k^{-1}v_k] \in \mathbb{R}^{n\times k}$. Note that when the preconditioning is fixed, i.e., $W_i = W$, flexible Arnoldi reduces to standard right preconditioned Arnoldi (see (2.9)).

Flexible Golub–Kahan decomposition. The flexible Golub–Kahan decomposition of $A \in \mathbb{R}^{m\times n}$ has been recently introduced in [9] to solve ℓ_p-regularized least squares problems. Given A, b, and iteration-dependent right preconditioning matrices $(W_k^{-1})^2$, the partial factorizations

$$AZ_k = U_{k+1}M_k \quad \text{and} \quad A^TU_{k+1} = V_{k+1}S_{k+1} \tag{2.12}$$

are updated at iteration k (for $k \le \min\{m,n\}$). In the first equation of (2.12), $M_k \in \mathbb{R}^{(k+1)\times k}$ is upper Hessenberg, $U_{k+1} \in \mathbb{R}^{m\times(k+1)}$ has orthonormal columns

with $u_1 = b/\|b\|_2$, and $Z_k = [(W_1^{-1})^2 v_1, \ldots, (W_k^{-1})^2 v_k] \in \mathbb{R}^{n\times k}$. Moreover, $S_{k+1} \in \mathbb{R}^{(k+1)\times(k+1)}$ is upper triangular and $V_{k+1} \in \mathbb{R}^{n\times(k+1)}$ has orthonormal columns. Note that for fixed preconditioning, i.e., $W_i = W_k$, FLSQR with preconditioner $(W_k^{-1})^2$ reduces to right preconditioned LSQR, which is mathematically equivalent to CG applied to the normal equations with split preconditioner W_k^{-1}. Although this relation is not stressed in [9], it can be observed in the definition of the search space for preconditioned LSQR in (2.10). The cost of computing these partial factorizations is dominated by one matrix vector product with A and one matrix vector product with A^T per iteration.

Detailed computations to update the partial flexible Arnoldi and flexible Golub–Kahan decompositions at the kth iteration are reported below. Notationwise, $[\cdot]_{i,j}$ denotes the (i,j)th entry of a matrix, and the vectors v_i, u_i, and z_i denote the ith column of the matrices V_k, U_k, and Z_k, correspondingly.

Flexible Arnoldi update

1: $z_k = W_k^{-1} v_k$
2: $w = A z_k$
3: Compute $[H]_{i,k} = w^T v_i$ for $i = 1, \ldots, k$ and set $w = w - \sum_{i=1}^{k} [H]_{i,k} v_i$
4: Set $[H]_{k+1,k} = ||w||_2$ and, if $[H]_{k+1,k} \neq 0$, take $v_{k+1} = w/[H]_{k+1,k}$

Flexible Golub–Kahan update

1: $w = A^T u_k$
2: Compute $[S]_{i,k} = w^T v_i$ for $i = 1, \ldots, k-1$ and set $w = w - \sum_{i=1}^{k-1} [S]_{i,k} v_i$
3: Set $[S]_{k,k} = ||w||_2$ and, if $[S]_{k,k} \neq 0$, take $v_k = w/[S]_{k,k}$
4: $z_k = (W_k^{-1})^2 v_k$
5: $w = A z_k$
6: Compute $[M]_{i,k} = w^T u_i$ for $i = 1, \ldots, k$ and set $w = w - \sum_{i=1}^{k} [M]_{i,k} u_i$
7: Set $[M]_{k+1,k} = ||w||_2$ and, if $[M]_{k+1,k} \neq 0$, take $u_{k+1} = w/[M]_{k+1,k}$

Flexible methods to solve ℓ_p-regularized least square problems have already been used in [18, 9], where, at the kth iteration, the following projected problem is solved:

$$\bar{y}_k = \arg\min_{\bar{y}} \|A Z_k \bar{y} - b\|_2^2 + \lambda \|\bar{y}\|_2^2 \quad \text{so that} \quad x_k = Z_k \bar{y}_k \,. \tag{2.13}$$

Note that $\bar{y}_k$ corresponds to the coefficients of the solution of (1.2) (in standard form) in the basis given by the columns of Z_k, which span a flexible Krylov space of dimension k with iteration dependent preconditioner W_k^{-1} and $(W_k^{-1})^2$ for FGMRES and FLSQR, respectively, where $W_k = \widetilde{W}^{(p,\tau)}(x_{k-1})$. Although extensive numerical tests show that methods (2.13) are efficient and deliver excellent reconstructions when compared to other Krylov solvers and other state-of-the-art methods for (1.5), it should be noted that solving problem (2.13) is not equivalent to solving problem (2.5) projected onto an appropriate flexible Krylov subspace at the kth iteration. Indeed, assume that n iterations of a flexible algorithm (2.13) have been performed, so that $\mathcal{R}(Z_n) = \mathbb{R}^n$: in this situation expression (2.13) corresponds to the Tikhonov problem (1.2) in standard form associated to (1.1) (and not the modification of the ℓ_2-ℓ_p problem in (2.4)). In other words, the "first-regularize-then-project" approach is not equivalent to the "first-project-then-regularize" approach for flexible Krylov solvers. Alternatively, this mismatch can be explained using the fact that, unlike in

the case of (nonflexible) preconditioned Krylov methods, in the problem projected using flexible Krylov subspaces there is no straightforward way of representing the variable $\bar{x}$ in (2.6) before "back-transformation." Note that [9] proposes to replace the regularization term $\|\bar{y}\|_2^2$ in (2.13) by $\|Z_k\bar{y}\|_2^2$: while (2.13) can be regarded as a hybrid regularization method that imposes additional standard form Tikhonov regularization on the projected solution $\bar{y}_k$, the regularization term $\|Z_k\bar{y}\|_2^2$ enforces standard form Tikhonov regularization on $x_k = Z_k\bar{y}_k$ and does not lead to a scheme equivalent to the "first-regularize-then-project" one, either.

In the following section, two algorithms exploiting flexible Krylov subspaces in connection with the "first-regularize-then-project" framework will be presented along with a proof of convergence of the resulting schemes.

3. Iteratively reweighted flexible Krylov subspace methods. In this section, two new algorithms are presented to solve (2.4) using a sequence of approximate problems of the form (2.5) and flexible Krylov subspaces (based on the flexible Arnoldi decomposition and the flexible Golub–Kahan decomposition, respectively).

Here and in the following, without loss of generality, no initial guess is considered for the solution of (2.4) in a "warm start" fashion; however, a possible initial guess $x_0 \neq 0$ may be purely used to initialize the weights (2.3) at the very first iteration of the algorithm. The presented algorithms are assumed to be breakdown-free, i.e., at iteration $k \leq \min\{m, n\}$, the approximation subspace $\mathcal{R}(Z_k)$ for the solution has dimension k.

3.1. The new IRW-FGMRES and IRW-FLSQR methods. The kth iteration of the new IRW-FGMRES and IRW-FLSQR methods computes an approximate solution x_k belonging to the space spanned by the columns of the matrix Z_k appearing in (2.11) or (2.12), respectively. More precisely, problem (2.5) is solved partially (i.e., in the space spanned by the columns of Z_k) as a projected least squares problem of the form

$$\bar{y}_k = \arg\min_{\bar{y}} \|AZ_k\bar{y} - b\|_2^2 + \lambda\|W_kZ_k\bar{y}\|_2^2 \quad \text{so that} \quad x_k = Z_k\bar{y}_k\,. \tag{3.1}$$

Let

$$W_kZ_k = Q_kR_k, \quad \text{with} \quad Q_k \in \mathbb{R}^{n\times k}, \quad R_k \in \mathbb{R}^{k\times k}, \tag{3.2}$$

be the reduced QR factorization of the tall and skinny matrix W_kZ_k, which can be computed efficiently (see, for example, [13]). Then (3.1) is equivalent to

$$\bar{y}_k = \arg\min_{\bar{y}} \|\bar{H}_k\bar{y} - \|b\|_2e_1\|_2^2 + \lambda\|R_k\bar{y}\|_2^2\,, \quad \text{so that} \quad x_k = Z_k\bar{y}_k, \tag{3.3}$$

for IRW-GMRES, or

$$\bar{y}_k = \arg\min_{\bar{y}} \|M_k\bar{y} - \|b\|_2e_1\|_2^2 + \lambda\|R_k\bar{y}\|_2^2\,, \quad \text{so that} \quad x_k = Z_k\bar{y}_k, \tag{3.4}$$

for IRW-FLSQR. With a notation analogous to (2.13), $\bar{y}_k$ corresponds to the coefficients of the solution of (2.5) in the basis formed by the columns of Z_k, which span a flexible Krylov space of dimension k with iteration-dependent preconditioning W_k^{-1} for IRW-FGMRES and $(W_k^{-1})^2$ for IRW-FLSQR (where $W_k = \widetilde{W}^{(p,\tau)}(x_{k-1})$). After the approximate solution x_k to problem (3.1) has been computed, the weights $W_{k+1} = \widetilde{W}^{(p,\tau)}(x_k)$ are (immediately) updated to be used in the next IRW-FGMRES or IRW-FLSQR iteration.

Although (3.1) might seem a rather unnecessarily convoluted formulation, since a change of variables for the regularization term is done and undone (i.e., an initial transformation into standard form in (2.6) eventually leads to a Tikhonov problem in general form), formulation (3.1) provides two main advantages over (2.8) and other IRN strategies based on Krylov subspaces. First, the iteration-dependent regularization matrix W_k favorably affects the approximation subspace for the solution of problems of the form (2.5), i.e.,

$$x_k \in \mathcal{R}(Z_k) = \mathcal{R}\left(\left[W_1^{-1}v_1, \ldots, W_k^{-1}v_k\right]\right),$$

for a set of vectors v_i that depend on the choice of IRW-FGMRES or IRW-FLSQR; see also [9, 20]. Second, problem (3.1) can be interpreted as a projection of the kth full-dimensional Tikhonov problem (2.5) (i.e., in a "first-regularize-then-project" framework). As a consequence, it can be proven that the sequence of approximate solutions $\{x_k\}_{k\geq 1}$ computed by IRW-FGMRES or IRW-FLSQR converges to the solution of problem (2.4).

Remark 3.1. Note that, assuming $n \leq m$ in (1.1), the IRW-FGMRES and IRW-FLSQR methods can be extended to the case when the number of iterations exceeds n by considering

$$x_k = \begin{cases} \arg\min_{x\in\mathcal{R}(Z_k)} T_k(x) & \text{for} \quad k = 1, \ldots, n-1, \\ \arg\min_{x\in\mathbb{R}^n} T_k(x) & \text{for} \quad k = n, \ldots, \end{cases} \tag{3.5}$$

where $T_k(x)$ is defined in (2.5). Indeed, when $n \leq k$, an iteration of IRW-FGMRES or IRW-FLSQR corresponds to an IRN iteration for ℓ_p regularization (1.5), where the solution of each subproblem (2.5) is computed in a "direct" fashion because the approximation subspace for the solution coincides with $\mathbb{R}^n$. Note, however, that this situation is not expected to happen in practice for large-scale problems.

Remark 3.2. Some numerical instabilities might happen in generating W_kZ_k in the regularization term in (3.1) when applying the new IRW-FGMRES and IRW-FLSQR methods, due to division by almost zeros in the weights component. Section 4 presents an example where this happens and discusses two possible fixes that can be adopted at the implementation level to improve stability.

The new IRW-FGMRES and IRW-FLSQR methods are sketched in Algorithm 3.1.

If $k \ll \min\{m, n\}$, the computational cost of the kth iteration of Algorithm 3.1 is dominated by the computational cost of updating the factorizations (2.11) or (2.12). Indeed, for IRW-FGMRES and assuming that A is dense, computing matrix-vector

Algorithm 3.1. IRW-FGMRES and IRW-LSQR methods.

1: Input: A, b, p, $\tau > 0$, x_0
2: Initialize: $v_1 = b/||b||_2$ for IRW-FGMRES, $u_1 = b/||b||_2$ for IRW-FLSQR
3: If $x_0 \neq 0$ $W_1 = \widetilde{W}^{(p,\tau)}(x_0)$ else $W_1 = \mathrm{I}_n$
4: **for** $k = 1, \ldots,$ until a stopping criterion is satisfied **do**
5: Update (2.11) (for IRW-FGMRES) or (2.12) (for IRW-FLSQR)
6: Compute $\bar{y}_k$ in (3.3) (for IRW-FGMRES) or in (3.4) (for IRW-FLSQR)
7: Compute $x_k = Z_k\bar{y}_k$
8: Update the weights $W_{k+1} = \widetilde{W}^{(p,\tau)}(x_k)$
9: **end for**

products with A amounts to $O(mn)$ flops (but could be much less if A is sparse or has some structure), while performing the orthonormalization steps amounts to $O(kn)$ flops. Forming the matrix $W_k Z_k$ and computing the QR factorization (3.2) amounts to $O(nk^2)$ flops, while solving problem (3.3) and forming x_k amounts to $O(k^3)$ flops. Similar estimates can be derived for IRW-FLSQR.

3.2. Convergence of IRW-FGMRES and IRW-FLSQR. Note that even if in practice IRW-FGMRES and IRW-FLSQR allow for an iteration-dependent choice of the regularization parameter λ in the functional $T^{(p,\tau)}(x)$ in (2.4), in this section λ is assumed to be known a priori and fixed throughout the iterations.

LEMMA 3.3. *Assume that no breakdown happens in the flexible Arnoldi and Golub–Kahan algorithms. Then the sequence* $\{T^{(p,\tau)}(x_k)\}_{k\geq 1}$ *for* $0 < p \leq 2$*, where* $T^{(p,\tau)}(x)$ *is defined in* (2.4)*, and where* x_k *is the approximate solution computed after* k *steps of the IRW-FGMRES or the IRW-FLSQR method, is decreasing monotonically and it is bounded from below by zero.*

Proof. Consider a fixed $p \in (0, 2]$ and $\tau > 0$. Since $T^{(p,\tau)}(x) \geq 0$, only the fact that $T^{(p,\tau)}(x_k)$ is monotonically decreasing needs to be proved, i.e., that $T^{(p,\tau)}(x_k) \leq T^{(p,\tau)}(x_{k-1})$ for every $k \geq 1$. Consider $T_k(x)$ defined in (2.5) (note that it is defined with respect to $W_k = \widetilde{W}^{(p,\tau)}(x_{k-1})$) and recall that $T_k(x)$ is a quadratic tangent majorant of $T^{(p,\tau)}(x)$ at point x_{k-1}, i.e.,

$$T^{(p,\tau)}(x_{k-1}) = T_k(x_{k-1}) \quad \text{and} \quad T^{(p,\tau)}(x) \leq T_k(x) \quad \forall x. \tag{3.6}$$

In particular, for x_k,

$$T^{(p,\tau)}(x_k) \leq T_k(x_k). \tag{3.7}$$

Moreover, recalling the definition of x_k in (3.1), and since $x_{k-1} \in \mathcal{R}(Z_{k-1}) \subset \mathcal{R}(Z_k)$,

$$T_k(x_k) = \min_{x \in \mathcal{R}(Z_k)} T_k(x) \leq T_k(x_{k-1}), \tag{3.8}$$

so, combining (3.6), (3.7), and (3.8),

$$T^{(p,\tau)}(x_k) \leq T_k(x_k) \leq T_k(x_{k-1}) = T^{(p,\tau)}(x_{k-1})\,, \tag{3.9}$$

which concludes the proof. ∎

THEOREM 3.4. *Under the same assumptions of Lemma* 3.3*, the sequence* $\{x_k\}_{k\geq 1}$*, where* x_k *is the approximated solution computed after* k *steps of IRW-FGMRES or IRW-FLSQR with* $p > 0$*, is such that*

$$\lim_{k\to\infty} \|x_k - x_{k-1}\|_2 = 0.$$

Moreover, it converges to a stationary point of $T^{(p,\tau)}$ *and, if* $p \geq 1$*, this is the unique solution of* (2.4)*.*

Proof. Thanks to Lemma 3.3, $\{T^{(p,\tau)}(x_k)\}_{k\geq 1}$ has a stationary point. The convergence result for $\{x_k\}_{k\geq 1}$ proved in Theorem 5 of [27] for majorization-minimization methods based on generalized Krylov subspaces, when $k \geq n$, can be applied in this setting as the same majorization for $T^{(p,\tau)}$ is used. ∎

It should be stressed that, although the regularization parameter λ in (3.1) is assumed fixed, the IRW-FGMRES and the IRW-FLSQR methods naturally allow for an iteration-dependent regularization parameter λ_k to be adaptively set at the kth iteration (e.g., at line 6 of Algorithm 3.1). Indeed, when considering inner-outer iterative schemes for (2.6) or flexible Krylov methods for (2.13), one can employ approaches typically used for hybrid methods (e.g., projected versions or approximations of well-known regularization parameter rules for Tikhonov problem (1.2); see [9, 18]). For IRW-FGMRES and IRW-FLSQR to be consistent with the "first-regularize-then-project" framework, one should make sure that the parameter λ_k selected at the kth iteration according to the adopted rule is a suitable λ for problem (2.5) and, eventually, for problem (1.5): although for projection methods based on standard Krylov subspaces convergence of λ_k to a λ can be guaranteed in some situations (e.g., when using standard Golub–Kahan bidiagonalization and the discrepancy principle; see [21]), it is not immediate to generalize these results to IRW-FGMRES and IRW-FLSQR. In the numerical experiments displayed in section 4 the discrepancy principle is employed to select the regularization parameter at each IRW-FGMRES or IRW-FLSQR iteration.

3.3. Alternative interpretation of IRW flexible methods. Augmented Krylov subspaces are most commonly used to incorporate an initial "guess" subspace of moderate dimension within a (traditional) Krylov subspace for the approximation of the solution of a linear system. In the framework of ill-posed problems, this approach is extremely beneficial if the initial "guess" vectors are chosen to model known features of the solution (see, e.g., [1, 2, 3, 15]); a combination of Tikhonov regularization and projection onto augmented Krylov subspaces has been considered in [24]. When performing iteratively reweighted schemes, a sequence of different but closely related problems of the form (2.5) or, equivalently, (2.6) is considered. Potentially, an augmented Krylov subspace method could be used to solve each of the problems if one had a good initial set of "guess" vectors. In this setting it is argued that IRW flexible Krylov methods can be regarded as particular instances of augmented Krylov methods where, when approximating the solution of the kth problem of the form (2.5) (i.e., at iteration $k \leq \min\{m, n\}$), the initial "guess" subspace is taken to be $\mathcal{R}(Z_{k-1})$ (i.e., the flexible Krylov subspace available from the previous iteration) and only one iteration of a (standard) Krylov method is performed (so that, in particular, the size of the augmentation subspace for the kth problem of the form (2.5) is $k-1$). This interpretation also draws similarities with the idea of recycling Krylov methods for sequences of linear systems [29, 38] and can be extended to flexible Krylov methods in general. Indeed, some analogies between flexible GMRES and augmented GMRES were already established in [8, 41]. Although the following derivations are specified for IRW-FGMRES and for augmented methods based on GMRES, they can be easily extended to handle IRW-FLSQR and augmented methods based on LSQR.

Consider the kth IRW-FGMRES iteration. Using the identity

$$Z_k = [Z_{k-1}, W_k^{-1} v_k] = W_k^{-1} [W_k Z_{k-1}, v_k] \,,$$

the flexible Arnoldi partial factorization (2.11) can be reformulated as

$$A[Z_{k-1}, W_k^{-1} v_k] = A W_k^{-1} [W_k Z_{k-1}, v_k] = [V_k, v_{k+1}] \bar{H}_k, \tag{3.10}$$

and the kth minimization problem (3.1), solved at the kth iteration of IRW-FGMRES, can be expressed as

$$\bar{y}_k = \arg\min_{\bar{y}} \| A W_k^{-1} [W_k Z_{k-1}, v_k] \bar{y} - b \|_2^2 + \lambda \| [W_k Z_{k-1}, v_k] \bar{y} \|_2^2 \,. \tag{3.11}$$

Then, $\bar{x}_k = [W_k Z_{k-1}, v_k]\bar{y}_k$ is an approximate solution of the kth problem of the form (2.6) that belongs to the space $\mathcal{R}([W_k Z_{k-1}, v_k])$, and $x_k = W_k^{-1}\bar{x}_k$ is an approximate solution of the kth problem of the form (2.5) that belongs to the space $\mathcal{R}(Z_k)$.

Now consider a single step of the augmented Arnoldi process with augmentation space $\mathcal{R}(Z_{k-1})$ and with starting vector

(3.12)
$$\hat{v}_k = \left(I - V_{k-1}V_{k-1}^T\right) r_{k-1} / \left\| \left(I - V_{k-1}V_{k-1}^T\right) r_{k-1} \right\|_2 , \quad \text{with} \quad r_{k-1} = b - Ax_{k-1} ,$$

so that $\hat{v}_k = v_k$. This leads to an approximation subspace for the solution of dimension k and can be written as follows:

1: Define $\hat{v}_k$ as in (3.12) and set $V_k = [V_{k-1}, \hat{v}_k]$.
2: Compute $\hat{z}_k = W_k^{-1}\hat{v}_k$.
3: Compute $\hat{w} = (I - V_k V_k^T)A\hat{z}_k$.
4: Take $[\hat{H}]_{k+1,k} = \|\hat{w}\|_2$.
5: Compute $\hat{v}_{k+1} = \hat{w}/[\hat{H}]_{k+1,k}$.

In the above algorithm, the matrix V_k in line 1 coincides with the matrix V_k in (3.10) because $\hat{v}_k = v_k$. Lines 3 to 5 can be rearranged as

$$[\hat{H}]_{k+1,k}\,\hat{v}_{k+1} = \left(I - V_k V_k^T\right) A\hat{z}_k \quad \text{so that} \quad A\hat{z}_k = V_k \left(V_k^T A\hat{z}_k\right) + \hat{v}_{k+1}[\hat{H}]_{k+1,k} .$$

Incorporating augmentation and considering the partial factorization (2.11) with k replaced by $k-1$, the following decomposition is obtained:

$$A[Z_{k-1}, \hat{z}_k] = [V_k, \hat{v}_{k+1}] \begin{bmatrix} \bar{H}_{k-1} & V_k^T A\hat{z}_k \\ 0 & [\hat{H}]_{k+1,k} \end{bmatrix} = [V_k, \hat{v}_{k+1}]\hat{H}_k. \tag{3.13}$$

Comparing the above algorithm to the flexible Arnoldi algorithm in section 2, it is immediate to see that $\hat{z}_k = W_k^{-1}\hat{v}_k = W_k^{-1}v_k = z_k$, and $\hat{v}_{k+1} = v_{k+1}$. Therefore, by inspection, it can be seen that this formulation is equivalent to (3.10) and that $\bar{H}_k = \hat{H}_k$.

As a consequence, the projection step performed to compute $\bar{y}_k$ in (3.11) using either the flexible or the augmented approach is equivalent, so the same kth approximate solution x_k of (3.1) is obtained.

The augmented method (3.13) mainly differs from the available augmented methods in the starting vector that is chosen for building the (standard) Krylov subspace: indeed, the latter take either the normalized right-hand side b (i.e., the (standard) Krylov subspace is built first, and then enriched with the initial "guess" subspace; see [15, 24]) or the orthogonal projection of b on the orthogonal complement of the initial "guess" subspace (i.e., the (standard) Krylov subspace is built preserving orthogonality to the initial "guess" subspace; see [1, 2, 3]). Note that the choice of the initial vector (3.12) for IRW-FGMRES more radically stems from the fact that $(I - V_k V_k^T)b = 0$, as $b \in \mathcal{R}(V_k)$.

The decomposition (3.13) associated to IRW-FGMRES is also analogous to the decompositions typically associated to recycling methods [38], the only difference being in the way the solution is computed (recycling often considers "warm restarts," where computing the solution at the kth iteration amounts to computing the correction of an initial guess).

4. Numerical experiments. In this section the results of three experiments concerned with imaging problems are presented to illustrate the behavior of the new methods. In all the experiments, x is the vector obtained by stacking the columns of a

two-dimensional discrete image. The new IRW-FGMRES and IRW-FLSQR methods are compared with other state-of-the-art solvers for (1.5) with $0 < p \leq 2$, including other solvers based on generalized and flexible Krylov methods, first-order optimization methods or optimization methods based on quadratic separable approximations of part of the objective function, and solvers that employ standard or preconditioned Krylov methods based on the Arnoldi and the Golub–Kahan bidiagonalization algorithms. To the best of our knowledge, comparisons between methods based on flexible and generalized Krylov subspaces have never been considered before. Table 1 summarizes the methods considered in this section, providing acronyms and brief descriptions thereof. Note that for all the considered examples, the computation of matrix-vector products with A and, possibly, A^T dominates the computational cost of each iteration of all the methods listed in Table 1. In particular, Krylov methods based on the (flexible) Golub–Kahan algorithm (i.e., IRW-FLSQR, IRN-hLSQR, (hybrid) FLSQR) have the same computational cost per iteration as GKSpq, FISTA, and SpaRSA, since they require one matrix-vector product with A and A^T; Krylov methods based on the (flexible) Arnoldi algorithm (i.e., IRW-FGMRES, IRN-hGMRES, (hybrid) FGMRES) are the ones with the lowest cost per iteration, since they require only one matrix-vector product with A. As a consequence, in the following tests, methods that require fewer iterations to compute solutions of comparable qualities have to be regarded as more efficient.

TABLE 1

Summary of the methods considered in this section for approximating the solution of problem (1.5).

Method	Description	Note	Reference	Marker
IRW-FGMRES IRW-FLSQR	the new Algorithm 3.1	adaptive reg. parameter selection	–	blue line
IRN-hGMRES IRN-hLSQR	IRN strategy within an inner-outer scheme	preconditioned hybrid GMRES or LSQR is used to solve (2.6) at each outer iteration; adaptive reg. parameter selection	[39]	green line
hybrid FGMRES hybrid FLSQR	hybrid versions of FGMRES or FLSQR	standard form Tikhonov regularization applied on the projected solution; adaptive reg. parameter selection	[9, 18]	pink line
FGMRES FLSQR	flexible GMRES or LSQR with sparsity-enforcing iteration-dependent preconditioning	no Tikhonov regularization for the projected problem	[9, 18]	dark red line
GKSpq	generalized Krylov subspace methods	initial subspace $\mathcal{K}_l(A^TA, A^Tb)$ with $l = 5$; adaptive reg. parameter selection	[31]	light blue line
FISTA	fast ISTA	accelerated first-order optimization method	[4]	purple line
SpaRSA	sparse reconstruction by separable approximation	quadratic separable approximations of part of the objective function	[46]	orange line

When a method allows the regularization parameter λ to be adaptively set at each iteration, this is done according to the discrepancy principle [34] as described below. Assuming that a good approximation of the 2-norm of the noise vector e appearing in (1.1) is available, a zero-finder is employed to solve the following nonlinear equation with respect to $\lambda \geq 0$ at the kth iteration:

$$\|Ax_k(\lambda) - b\|_2 = \eta \|e\|_2 \,, \tag{4.1}$$

where $x_k(\lambda)$ is the approximate solution at iteration k given as a function of the regularization parameter λ, and $\eta \geq 1$ is a safety parameter. Note that (4.1) is guaranteed to have a solution as soon as $\|Ax_k(0) - b\|_2 \leq \|e\|_2$. For IRW-FGMRES,

$$\begin{aligned} x_k(\lambda) = Z_k \bar{y}_k &= Z_k \left(\bar{H}_k^T \bar{H}_k + \lambda R_k^T R_k \right)^{-1} \bar{H}_k^T \|b\|_2 e_1 \\ &= Z_k \left(\bar{H}_k^T \bar{H}_k + \lambda R_k^T R_k \right)^{-1} \bar{H}_k^T V_{k+1}^T b, \end{aligned} \tag{4.2}$$

where $\bar{H}_k$ is defined in (2.11) and R_k is obtained computing the reduced QR factorization of $W_k Z_k$; see (3.2). Then

$$\begin{aligned} \|Ax_k(\lambda) - b\|_2 &= \left\| AZ_k \left(\bar{H}_k^T \bar{H}_k + \lambda R_k R_k^T \right)^{-1} \bar{H}_k^T V_{k+1}^T b - b \right\|_2 \\ &= \left\| V_{k+1} \bar{H}_k \left(\bar{H}_k^T \bar{H}_k + \lambda R_k R_k^T \right)^{-1} \bar{H}_k^T V_{k+1}^T b - b \right\|_2 \\ &= \left\| \bar{H}_k \left(\bar{H}_k^T \bar{H}_k + \lambda R_k R_k^T \right)^{-1} \bar{H}_k^T \|b\|_2 e_1 - \|b\|_2 e_1 \right\|_2 , \end{aligned} \tag{4.3}$$

so that applying the discrepancy principle (4.1) does not require performing any additional matrix-vector product with A per iteration. An analogous argument can be made specifically for IRW-FLSQR (as expression (4.3) formally holds for IRW-FLSQR after replacing the matrix $\bar{H}_k$ by M_k), as well as for most of the algorithms listed above; see also [30, 19]. Note that although synthetic noise e with known $\|e\|_2$ is always used in the following, estimates of the noise level or alternative parameter choice strategies that do not require an estimate of $\|e\|_2$ can be used if $\|e\|_2$ is not immediately available; see, e.g., [21, 45]. When no adaptive regularization parameter choice is supported (e.g., for FISTA and SpaRSA), the value of the regularization parameter computed by IRW-FGMRES or IRW-FLSQR (upon iteration termination) is used. Alternatively, such solvers can be run from scratch for different preselected values of the regularization parameter and the best solution can be picked according to some criterion, resulting in a very computationally demanding strategy.

Throughout all the experiments, if not stated otherwise, the values $p = 1$ and $\tau = 10^{-10}$ are chosen in (2.3), $\eta = 1$ is chosen in (4.1), and all the solvers are set to perform 200 (total) iterations. Although, provided that a suitable value of the regularization parameter is set at each iteration, the quality of the reconstructions computed by the new methods does not significantly deteriorate as the iterations proceed, one or more stopping criteria should be set in practice. A reasonable choice is to stop at the first iteration k such that

$$\frac{|\lambda_k - \lambda_{k-1}|}{\lambda_k} < \theta_1 \quad \text{or} \quad \frac{|s(x_k) - s(x_{k-1})|}{s(x_k)} < \theta_2, \tag{4.4}$$

where $\theta_1, \theta_2 > 0$ are user-selected thresholds, and where $s(\cdot)$ is a (practical) measure of the sparsity of the solution. In the following, given a vector y,

$$s(y) = \#\left\{ i \,:\, |[y]_i| \geq 10^{-3} ||y||_2 \right\}, \quad \text{where } \# \text{ denotes cardinality.} \tag{4.5}$$

Stopping criteria (4.4) monitor the stabilization of some relevant quantities for the solution, so that one can expect x_k not to vary too much once they are satisfied; see [19]. In all the graphs presented below, the iteration satisfying the first stopping criterion in (4.4) with $\theta_1 = 10^{-4}$ is marked by a circle, and the iteration satisfying the second stopping criterion in (4.4) with $\theta_2 = 10^{-10}$ is marked by a triangle.

Experiment 1. The first experiment is concerned with image deblurring. The `star_cluster` test problem from *Restore Tools* [35] is used to generate an exact test image of size 256×256 pixels (so $n = 65536$ in (1.1)) and a square blurring matrix modeling *spatially variant blur* (we refer interested readers to [36] for a discussion of how the matrix A is represented and how matrix-vector products can be done efficiently). The measurements are corrupted by Gaussian white noise e of level $\|e\|_2/\|b_{true}\|_2 = 10^{-2}$. The setting for this example can be observed in Figure 1. Note that $s(x_{true}) = 470$, i.e., only approximately 0.07% of the pixels can be regarded as different from zero in practice, according to definition (4.5). This example has been mimicked from [18]. Since A is square, the performance of IRW-FGMRES can be tested.

Figure 2 displays the behavior of the relative errors versus the number of iterations for the methods listed in Table 1. It can be observed in Figure 2(a) that IRW-FGMRES shows a faster and more stable convergence when compared to other standard methods for ℓ_2-ℓ_p regularization. In particular, the new method stabilizes to roughly the same value of the relative error as IRN and FISTA, while SpaRSA converges to a reconstruction of worse quality. Even restricting the comparisons to other methods that build only one generalized or flexible Krylov subspace for the solution, the new IRW-FGMRES method shows a more desirable behavior. Indeed, it can be observed in Figure 2(b) that the solver based on FGMRES displays some semiconvergence; this feature is shared by the hybrid version of FGMRES and may appear because a Tikhonov problem in standard form is solved, so that sparsity is only enforced through the construction of a suitable flexible Krylov subspace. Also, within the maximum number of allowed iterations, the quality of the solution computed by the solver based on generalized Krylov subspaces is lower than the IRW-FGMRES one: this shows that, for this test problem, the approximation subspace for the solution computed by IRW-FGMRES is better than the one computed by GKSpq.

Figure 3(a) displays the values of the relative residuals $\|b - Ax_k(\lambda)\|_2/\|b\|_2$ versus the number of iterations k. One can clearly see that, since λ is adaptively set at each

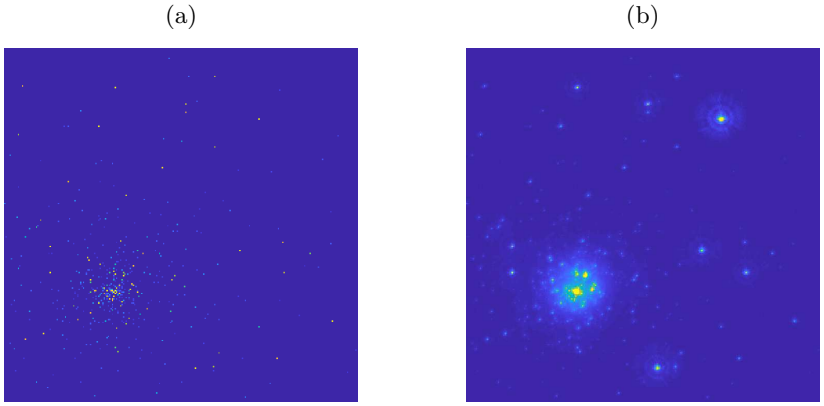


FIG. 1. *Experiment* 1. *Setting for the* `star_cluster` *test problem.* (a) *True image* x_{true}. (b) *Noisy measurement* b.

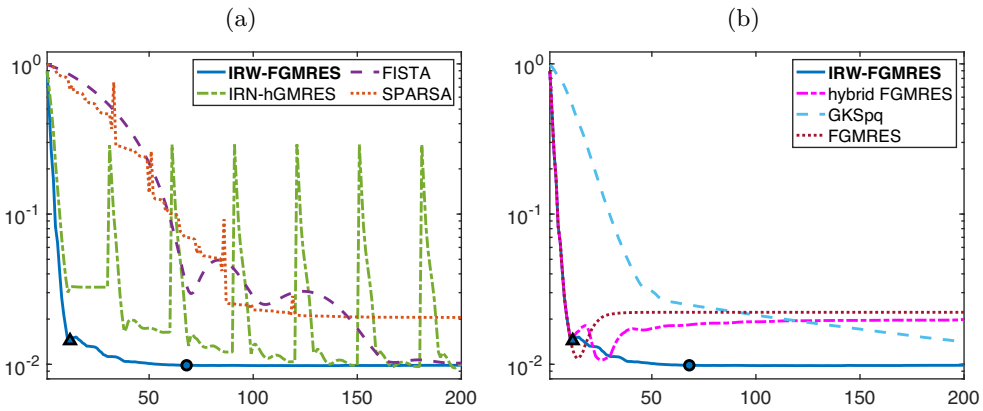


FIG. 2. *Experiment* 1. *History of relative error norms (i.e.,* $\|x_k(\lambda) - x_{true}\|_2/\|x_{true}\|_2$ *against iteration number* k*) for the new IRW-FGMRES, compared to* (a) *other standard solvers for the* ℓ_2-ℓ_1 *problem,* (b) *other flexible and generalized Krylov-based solvers. The circle and triangle markers correspond to stopping criteria* (4.4) *based on the stabilization of* λ *and* $s(x_k)$*, respectively.*

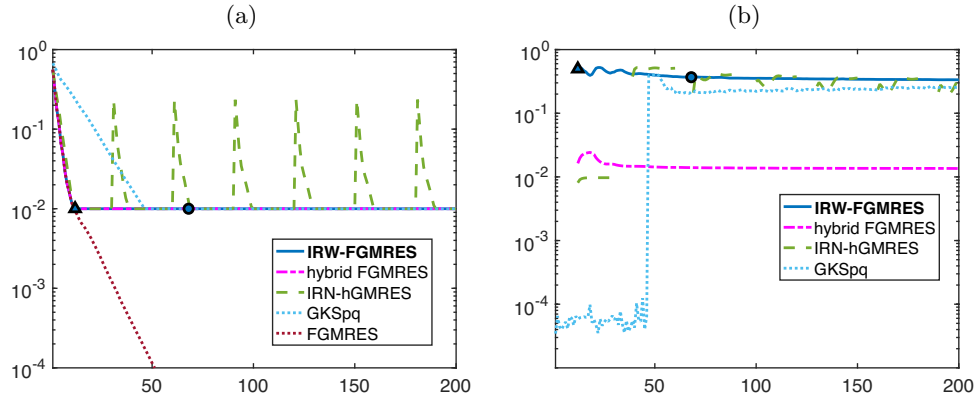


FIG. 3. *Experiment* 1. *Methods based on Krylov subspaces.* (a) *History of the relative residuals.* (b) *History of the regularization parameters. The circle and triangle markers correspond to stopping criteria* (4.4) *based on the stabilization of* λ *and* $s(x_k)$*, respectively.*

iteration using the discrepancy principle (for all the displayed methods except for FGMRES), the relative residual eventually stabilizes around the noise level, as should happen for regularization methods applied to ill-posed problems: this happens quite quickly for methods based on the flexible Arnoldi algorithm, but sensibly later for the GKSpq method (coherently to what is observed in Figure 2(a)). Figure 3(b) displays the values of the regularization parameters $\lambda = \lambda_k$ selected at each iteration versus the number of iterations k. It can be observed that the regularization parameter chosen by the new IRW-FGMRES method quickly stabilizes to a value that is similar to the one eventually selected by the IRN and GKSpq methods. The regularization parameter chosen by the hybrid version of FGMRES stabilizes to a different value, which is more similar to the one selected during the first IRN outer iteration, i.e., when a Tikhonov problem in standard form is solved. This behavior is consistent with the arguments presented in sections 2 and 3. Indeed, similarly to IRN and GKSpq, IRW-FGMRES can be proved to converge to a stationary point of (2.4): therefore it should be expected that the regularization parameter adaptively selected by these methods

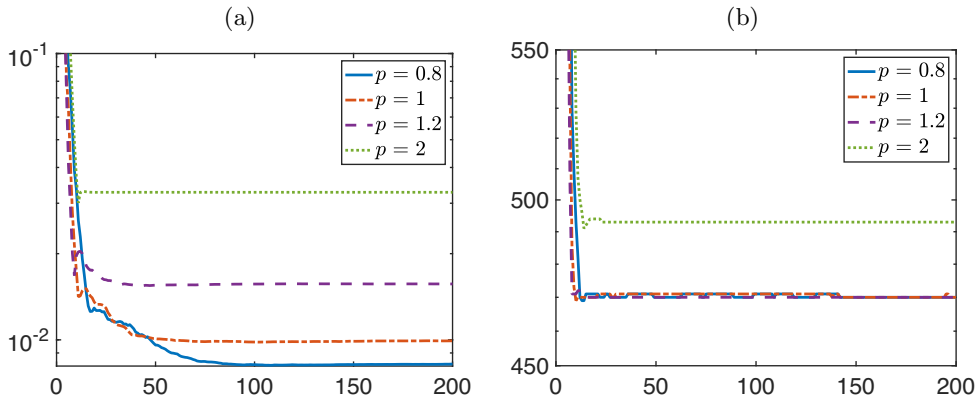


FIG. 4. *Experiment* 1. (a) *History of the IRW-FGMRES relative error norms for different values of p in the ℓ_p regularization term.* (b) *History of $s(x_k)$ for IRW-FGMRES and for different values of p in the ℓ_p regularization term.*

according to the discrepancy principle also stabilizes around a common value. On the contrary, hybrid FGMRES imposes additional standard form Tikhonov regularization on the projected solution: therefore it should be expected that the regularization parameter stabilizes around a value suitable for standard form Tikhonov regularization.

Finally, Figure 4(a) displays the history of relative errors obtained using IRW-FGMRES for different values of p in the ℓ_p regularization term. Note that since the quality of the solution generally improves when taking $p < 1$ (coherently with the fact that x_{true} is very sparse), one can expect that IRN-FGMRES is converging to a global minimum when started with $x_0 = 0$ for this test problem. Correspondingly, Figure 4(b) displays the values of $s(x_k)$ versus the number of iterations k. It can be observed that when the value of p in the ℓ_p regularization term is 2, the recovered solution is considerably less sparse than x_{true}, whereas for smaller values of p, the value of $s(x_k)$ approximates $s(x_{true}) = 470$. In particular, note that when $p = 1$, $s(x_k)$ converges to $s(x_{true}) = 470$ when using IRW-FGMRES. Even if not shown, this is also true for FISTA, SpaRSA, IRN-hGMRES, FGMRES, and hybrid FGMRES. Similarly, the solution obtained using the GKSpq method at the end of the iterations had an $s(x_k)$ of 472.

Experiment 2. The second test problem uses the so-called `hst` (Hubble space telescope) test image together with the spatially invariant `speckle medium blur` linear operator available within *IR Tools* [17]. The noise level is $\|e\|_2/\|b_{true}\|_2 = 10^{-2}$ and $\eta = 1$ is chosen in (4.1). The setting for this experiment can be observed in Figure 5. The object displayed in this test image is not as sparse as in the previous test problem; the overall sparsity is associated to the uniform (zero) background. Note that, in this example, the square matrix $A \in \mathbb{R}^{n\times n}$ (where $n = 65536$) is generated by a highly anisotropic blur (see Figure 5(b)): in this situation, there is no guarantee that GMRES can perform well; see [14]. For this reason, only the performance of methods based on LSQR will be compared.

The relative error history associated to different solvers for (2.4) is displayed in Figure 6. It should be stressed that, when running IRW-FLSQR for this experiment, $\tau = 0.01$ is set in (2.3) to avoid numerical instabilities happening in the generation of $W_k Z_k$ (as mentioned in Remark 3.2). As can be seen in Figure 7(a), a smaller value of τ would lead to solutions of worse quality. Alternatively, Figure 7(b) shows the history of the relative errors when the components of the weights $W_k = \widetilde{W}^{(p,\tau)}(x_{k-1,\star})$ are set to 0 in (2.5) if they are higher than a certain threshold τ_W (as suggested in [39]).

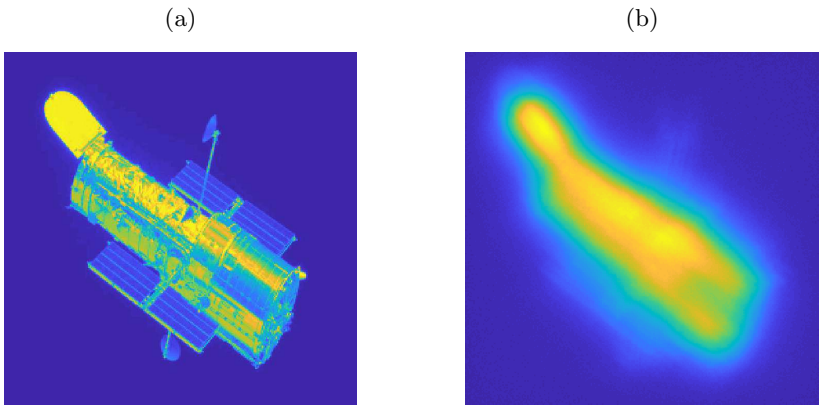


FIG. 5. *Experiment* 2. *Setting for the* `hst` *test problem.* (a) *True image* x_{true}. (b) *Noisy measurement* b.

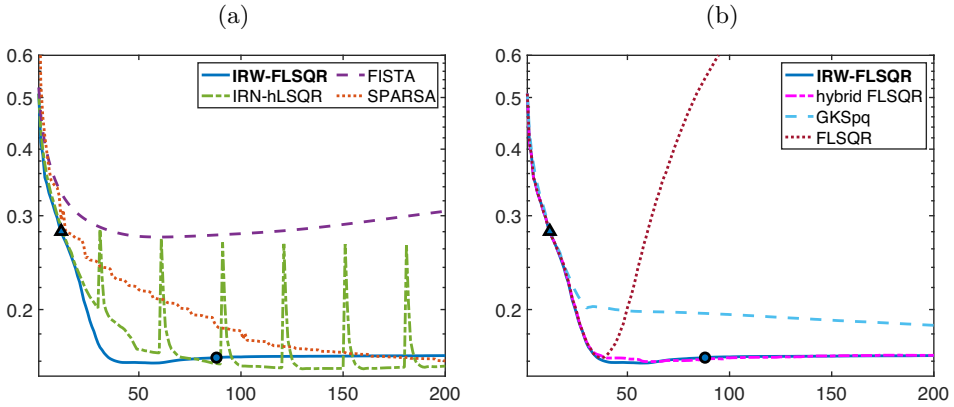
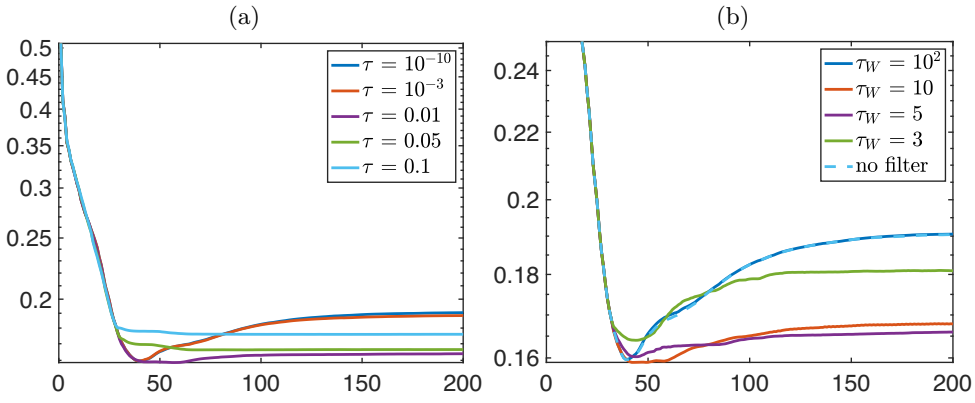


FIG. 6. *Experiment* 2. *History of relative error norms for the new IRW-FLSQR, compared to* (a) *other standard solvers for the* $\ell_2 - \ell_1$ *problem,* (b) *other flexible and generalized Krylov-based solvers. The circle and triangle markers correspond to stopping criteria* (4.4) *based on the stabilization of* λ *and* $s(x_k)$*, respectively.*

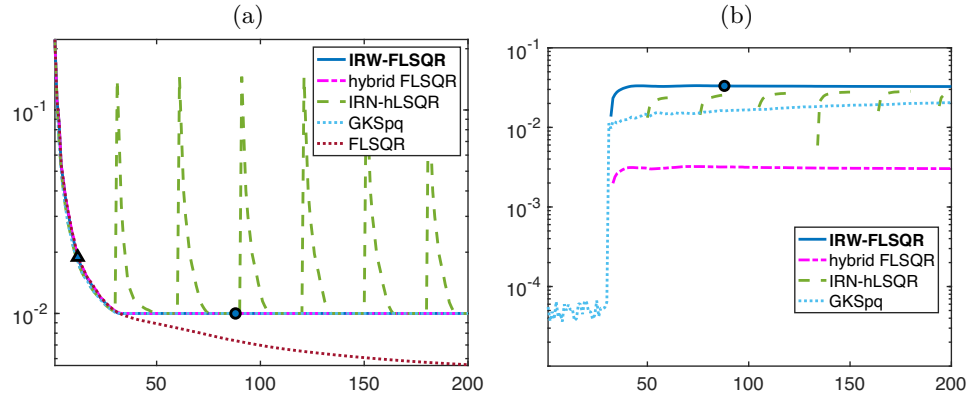
As in the previous example, Figure 8(a) displays the values of the relative residuals $\|b - Ax_k(\lambda)\|_2/\|b\|_2$ versus the number of iterations k, and Figure 8(b) displays the values of the regularization parameters $\lambda = \lambda_k$ selected at each iteration k according to the discrepancy principle. The behavior of these quantities is very similar to that observed in the previous example and it can be interpreted in the same way.

Experiment 3. This test problem models sparse X-ray tomographic reconstruction with oversampled data. The chosen test phantom is the `ppower` image from [26], generated in such a way that only 10% of its pixels are exactly nonzero; this phantom is also fairly smooth (see Figure 9(a)). A measurement geometry consisting of 362 equidistant parallel beams rotated around 224 equidistant angles between 1° and 180° is considered. This corresponds to a discrete forward operator $A \in \mathbb{R}^{m\times n}$ with $m = 81088$ and $n = 65536$, so that only methods based on the Golub–Kahan decomposition can be compared. The noise level in this example is $\|e\|_2/\|b_{true}\|_2 = 1.5{\cdot}10^{-2}$.

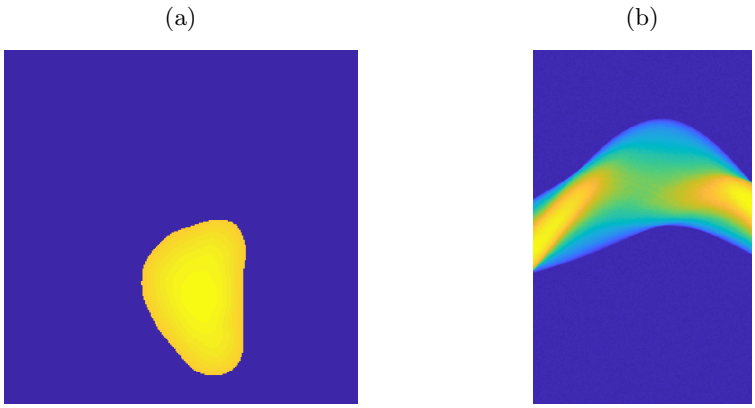
The convergence results for this tomography example with oversampled data are displayed in Figures 10 and 11. The methods based on flexible Krylov subspaces all



665 FIG. 7. *Experiment* 2. *Different strategies to stabilize the quality of the solution. History of the*
666 *relative error norms for the new IRW-FLSQR:* (a) *for different values of* τ, (b) *for different values*
667 *of* τ_W.



668 FIG. 8. *Experiment* 2. *Methods based on Krylov subspaces.* (a) *History of the relative residuals.*
669 (b) *History of the regularization parameters. The circle and triangle markers correspond to stopping*
670 *criteria* (4.4) *based on the stabilization of* λ *and* $s(x_k)$, *respectively.*



691 FIG. 9. *Experiment* 3. *Setting for the* `ppower` *test problem.* (a) *True phantom* x_{true}. (b) *Noisy*
692 *sinogram measurement* b.

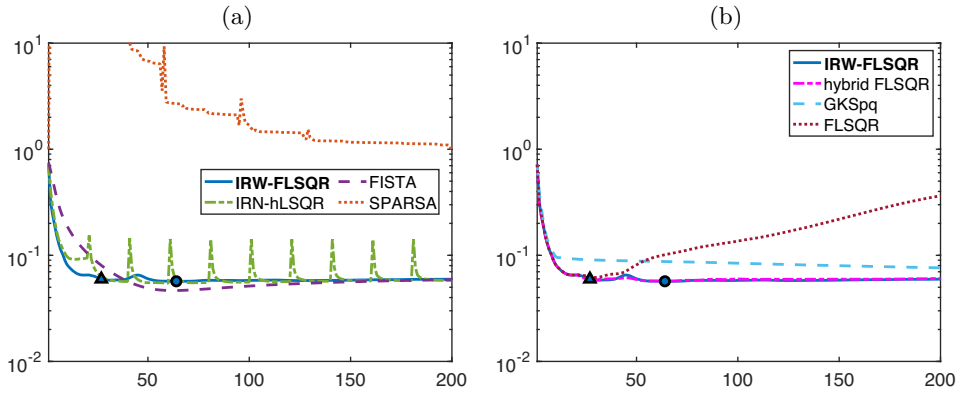


FIG. 10. *Experiment* 3. *History of relative error norms for the new IRW-FLSQR, compared to* (a) *other standard solvers for the* $\ell_2 - \ell_1$ *problem;* (b) *other flexible and generalized Krylov-based solvers. The circle and triangle markers correspond to stopping criteria* (4.4) *based on the stabilization of* λ *and* $s(x_k)$*, respectively.*

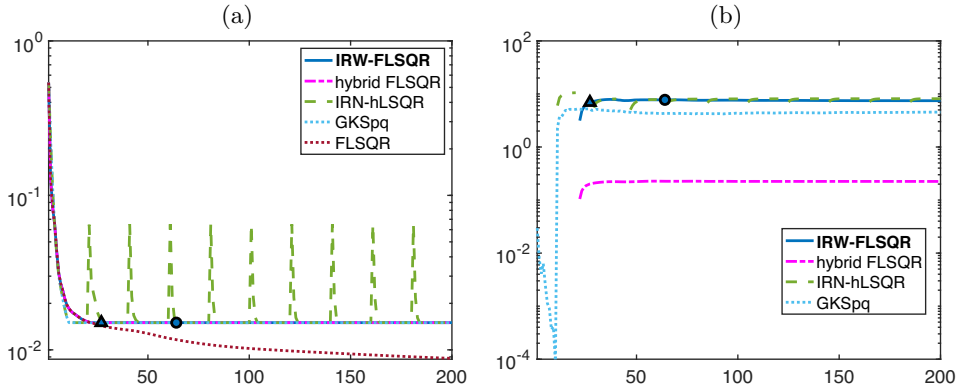


FIG. 11. *Experiment* 3. *Methods based on Krylov subspaces.* (a) *History of the relative residuals.* (b) *History of the regularization parameters chosen according to the discrepancy principle. The circle and triangle markers correspond to stopping criteria* (4.4) *based on the stabilization of* λ *and* $s(x_k)$*, respectively.*

perform similarly well. FISTA seems to deliver a solution of slightly better quality than IRW-FLSQR, but it takes more iterations to do so. SpaRSA seems to perform poorly for this test problem; it may be expected that experimenting with different values of the regularization parameter could lead to an improved solution.

5. Conclusions. This paper presents two new algorithms, called IRW-FGMRES and IRW-FLSQR, that efficiently solve the ℓ_2-ℓ_p minimization problem (1.5) by partially solving a sequence of quadratic problems arising from the IRN strategy. The new methods compute approximate solutions belonging to flexible Krylov subspaces of increasing dimension that encode regularization through iteration-dependent "preconditioning" so as to avoid nested loops of iterations and build only one approximation subspace for the solution. With respect to other available IRN solvers, the new approach not only improves the efficiency of the algorithm but also avoids the need of choosing stopping criteria for the inner iterations. Moreover, the regularization parameter can be set adaptively along the iterations (even using strategies other than

the discrepancy principle, which is considered in this paper). The new flexible Krylov solvers are supported by a solid theoretical justification: indeed, the sequence of approximate solutions given by Algorithm 3.1 is guaranteed to converge to the solution of the smoothed formulation (2.4) of problem (1.5).

Extensive numerical testing, involving large-scale inverse problems in imaging, shows that IRW-FGMRES and IRW-FLSQR are competitive with other standard implementations of IRN methods as well as other optimization methods. Moreover, although IRW-FGMRES can only be applied to a square coefficient matrix A and is not guaranteed to work well if A is highly nonnormal, it requires only a single matrix-vector product with A at each iteration, while IRW-FLSQR needs an additional matrix-vector product with A^T at each iteration. It is worth highlighting again that, although the hybrid implementations of FGMRES, FLSQR [18, 9] and IRW-FGMRES, IRW-FLSQR have a similar behavior in most of the performed numerical tests, the former still lack a solid theoretical justification of convergence.

Future work will include a theoretical investigation of the convergence of IRW-FGMRES and IRW-FLSQR in the presence of a variable regularization parameter that is automatically set at each iteration according to a given rule, and the extension of the new IRW flexible Krylov methods to handle more involved regularizers, such as total variation and generalizations thereof.

REFERENCES

[1] J. Baglama and L. Reichel, *Augmented GMRES-type methods*, J. Numer. Linear Algebra Appl., 14 (2007), pp. 337–350, https://doi.org/10.1002/nla.518.

[2] J. Baglama and L. Reichel, *Decomposition methods for large linear discrete ill-posed problems*, J. Comput. Appl. Math., 198 (2007), pp. 332–343, https://doi.org/10.1016/j.cam.2005.09.025.

[3] J. Baglama, L. Reichel, and D. Richmond, *An augmented LSQR method*, Numer. Algorithms, 64 (2013), https://doi.org/10.1007/s11075-012-9665-8.

[4] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[5] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996, https://doi.org/10.1137/1.9781611971484.

[6] A. Buccini, M. Donatelli, and L. Reichel, *Iterated Tikhonov regularization with a general penalty term*, Numer. Linear Algebra Appl., 24 (2017), e2089, https://doi.org/10.1002/nla.2089.

[7] D. Calvetti, *Preconditioned iterative methods for linear discrete ill-posed problems from a Bayesian inversion perspective*, J. Comput. Appl. Math., 198 (2007), pp. 378–395.

[8] A. Chapman and Y. Saad, *Deflated and augmented Krylov subspace techniques*, Numer. Linear Algebra Appl., 4 (1996), pp. 43–66, https://doi.org/10.1002/(SICI)1099-1506(199701/02)4:1⟨43::AID-NLA99⟩3.0.CO;2-Z.

[9] J. Chung and S. Gazzola, *Flexible Krylov methods for ℓ_p regularization*, SIAM J. Sci. Comput., 41 (2019), pp. S149–S171, https://doi.org/10.1137/18M1194456.

[10] J. Chung, M. E. Kilmer, and D. P. O'Leary, *A framework for regularization via operator approximation*, SIAM J. Sci. Comput., 37 (2015), pp. B332–59.

[11] J. Chung, J. G. Nagy, and D. P. O'Leary, *A weighted GCV method for Lanczos hybrid regularization*, Electron. Trans. Numer. Anal., 28 (2008), pp. 149–167.

[12] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, *Iteratively reweighted least squares minimization for sparse recovery*, Comm. Pure Appl. Math., 63 (2010), pp. 1–38, https://doi.org/10.1002/cpa.20303.

[13] J. Demmel, L. Grigori, M. Hoemmen, and J. Langou, *Communication-optimal parallel and sequential QR and LU factorizations*, SIAM J. Sci. Comput., 34 (2012), pp. A206–A239.

[14] M. Donatelli, A. Neuman, and L. Reichel, *Square regularization matrices for large linear discrete ill-posed problems*, Numer. Linear Algebra Appl., 19 (2012), pp. 896–913.

[15] Y. Dong, H. Garde and P. C. Hansen, *R3GMRES: Including prior information in GMRES-type methods for discrete inverse problems*, Electron. Trans. Numer. Anal., 42 (2014), pp. 136–146.

[16] M. Fornasier and H. Rauhut, *Compressive sensing*, in Handbook of Mathematical Methods in Imaging, Springer, New York, 2011, pp. 187–228, https://doi.org/10.1007/978-0-387-92920-0_6.
[17] S. Gazzola, P. Hansen, and J. G. Nagy, *IR Tools: A MATLAB package of iterative regularization methods and large-scale test problems*, Numer. Algorithms, 81 (2019), pp. 773–811, https://doi.org/10.1007/s11075-018-0570-7.
[18] S. Gazzola and J. Nagy, *Generalized Arnoldi-Tikhonov method for sparse reconstruction*, SIAM J. Sci. Comput., 36 (2014), B225–B247, https://doi.org/10.1137/130917673.
[19] S. Gazzola, P. Novati, and M. R. Russo, *On Krylov projection methods and Tikhonov regularization*, Electron. Trans. Numer. Anal., 44 (2015), pp. 83–123.
[20] S. Gazzola and M. Sabaté Landman, *Flexible GMRES for total variation regularization*, BIT, 59 (2019), pp. 721–746.
[21] S. Gazzola and M. Sabaté Landman, *Krylov Methods for Inverse Problems: Surveying classical, and introducing new, algorithmic approaches*, GAMM-Mitt., 43 (2020), https://doi.org/10.1002/GAMM.202000017.
[22] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343, https://doi.org/10.1137/080725891.
[23] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[24] P. Hansen, Y. Dong, and K. Abe, *Hybrid enriched bidiagonalization for discrete ill-posed problems*, Numer. Linear Algebra Appl., 26 (2019), e2230, https://doi.org/10.1002/nla.2230.
[25] P. C. Hansen, *Discrete Inverse Problems*, Fundam. Algorithms 7, SIAM, Philadelphia, 2010, https://doi.org/10.1137/1.9780898718836.
[26] P. C. Hansen and J. S. Jorgensen, *AIR Tools* II*: Algebraic iterative reconstruction methods, improved implementation*, Numer. Algorithms, 79 (2018), pp. 107–137, https://doi.org/10.1007/s11075-017-0430-x.
[27] G. Huang, A. Lanza, S. Morigi, L. Reichel, and F. Sgallari, *Majorization-minimization generalized Krylov subspace methods for ℓ_p-ℓ_q optimization applied to image restoration*, BIT, 57 (2017), pp. 351–378.
[28] Y. Huang and Z. Jia, *Some results on the regularization of LSQR for large-scale discrete ill-posed problems*, Sci. China Math., 60 (2017), pp. 701–718.
[29] M. E. Kilmer and E. de Sturler, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166.
[30] M. E. Kilmer and D. P. O'Leary, *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1204–1221, https://doi.org/10.1137/S0895479899345960.
[31] A. Lanza, S. Morigi, L. Reichel, and F. Sgallari, *A generalized Krylov subspace method for ℓ_p-ℓ_q minimization*, SIAM J. Sci. Comput., 37 (2015), pp. S30–S50.
[32] A. Lanza, S. Morigi, I. Selesnick, and F. Sgallari, *Nonconvex nonsmooth optimization via convex–nonconvex majorization–minimization*, Numer. Math., 136 (2016), pp. 343–381, https://doi.org/10.1007/s00211-016-0842-x.
[33] X. Li, D. Sun, and K. Toh, *A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems*, SIAM J. Optim., 28 (2018), pp. 433–458, https://doi.org/10.1137/16M1097572.
[34] V. A. Morozov, *On the solution of functional equations by the method of regularization*, Soviet Math. Dokl., 7 (1966), pp. 414–417.
[35] J. Nagy, K. Palmer, and L. Perrone, *Iterative methods for image deblurring: A MATLAB object-oriented approach*, Numer. Algorithms, 36 (2004), pp. 73–93.
[36] J. G. Nagy and D. P. O'Leary, *Restoring images degraded by spatially variant blur*, SIAM J. Sci. Comput., 19 (1998), pp. 1063–1082.
[37] D. P. O'Leary and J. A. Simmons, *A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 474–489.
[38] M. Parks, E. de Sturler, G. Mackey, D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2004), pp. 1651–1674, https://doi.org/10.1137/040607277.
[39] P. Rodriguez and B. Wohlberg, *An efficient algorithm for sparse representations with ℓ^p data fidelity term*, in Proceedings of the 4th IEEE Andean Technical Conference, 2008.
[40] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469, https://doi.org/10.1137/0914028.
[41] Y. Saad, *Analysis of augmented Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 18 (1995), pp. 435–449, https://doi.org/10.1137/S0895479895294289.

834 [42] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
835 [43] V. Simoncini and D. Szyld, *Flexible inner-outer Krylov subspace methods*, SIAM J. Numer.
836 Anal., 40 (2002), pp. 2219–2239, https://doi.org/10.1137/S0036142902401074.
837 [44] V. Simoncini and D. Szyld, *Recent computational developments in Krylov subspace*
838 *methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59,
839 https://doi.org/10.1002/nla.499.
840 [45] L. Tenorio, *An Introduction to Data Analysis and Uncertainty Quantifica-*
841 *tion for Inverse Problems*, Math. Ind. (Phila.) 3, SIAM, Philadelphia, 2017,
842 https://doi.org/10.1137/1.9781611974928.
843 [46] S. J. Wright, R. D. Nowak, and M. Figueiredo, *Sparse reconstruction by sep-*
844 *arable approximation*, IEEE Trans. Signal Process., 57 (2009), pp. 2479–2493,
845 https://doi.org/10.1109/icassp.2008.4518374.

3.3 Conclusions

This chapter studies the use of Kylov methods in combination with ℓ_2-ℓ_p regularization. This requires generalizing the Krylov subspace methods reviewed in Chapter 2 to incorporate iteration-dependent preconditioning, and this is done using flexible Krylov subspaces. Moreover, the particular reformulation of ℓ_2-ℓ_p regularization given in [GNL21] is used to give theoretical guarantees of convergence.

In the next chapter, flexible Krylov methods will be used in combination with TV regularization in a similar fashion to [GNL21]. However, the generalization of the algorithms in [GNL21] is not straightforward, as TV regularization as formulated in Chapter 4 of this thesis involves (partially) solving a sequence of Tikhonov problems with iteration-dependent non-square regularization matrices. The next chapter describes a new algorithm for TV regularization using flexible Krylov subspaces.

Chapter 4

Flexible GMRES for total variation regularization

This chapter presents a new algorithm based on total variation (TV) regularization for large-scale linear ill-posed problems. The new approach requires constructing a sequence of quadratic problems that are partially solved at each iteration using flexible GMRES (FGMRES). I present joint work with Silvia Gazzola, which is published in BIT Numerical Mathematics as open access publication [GL19].

4.1 Outline of the paper

This paper presents a new algorithm to solve large-scale linear ill-posed inverse problems using total variation regularization as defined in (1.21), which is approximated by a sequence of Tikhonov problems of the form (1.37). Unlike ℓ_2-ℓ_p regularization, presented in Chapter 3, the regularization matrix $W_k D$ associated to the quadratic approximation of the TV functional at iteration k (i.e., of the form (1.37)), is not square. This entails a more challenging situation when transforming (1.37) into standard form, which involves the computation of the A-weighted pseudoinverse of the matrix $W_k D$. Assuming that the original coefficient matrix A is square, the transformed problem is modified again to obtain a new equivalent problem with a square coefficient matrix, that is solved using flexible GMRES (FGMRES). Efficient implementation strategies are proposed to compute these transformations and are therefore included in [GL19, Section 4].

The performance of the new algorithm, TV-FGMRES, is illustrated on three deblurring problems (see Section 1.2). In particular, as already stated in Section 1.5.3, [GL19, Section 3] includes an illustration of how priorconditioning modifies the flexible Krylov subspace where the solution is sought for a simple one-dimensional deconvolution example.

Most of the content included in [GL19, Section 1] is already explained in Chapter 1 of this thesis, and can therefore be skipped by the reader. Similarly, most of the material mentioned in [GL19, Section 2] is summarized in Chapter 1 of this thesis, but is presented with more detail in [GL19, Section 2] and with a slightly different notation that is more tailored to TV regularization. [GL19, Section 2] also includes important remarks on the smoothing of the TV regularization functional. Therefore, we recommend reading it, although parts of it should be very familiar to the reader. There are some discrepancies between the notation in [GL19] and Chapter 1 of this thesis: these are summarized in Table 4.1.

Table 4.1: Notational discrepancies between this thesis and paper [GL19]

Discrepancies	Thesis	Paper
original linear problem	$Ax = b = b_{true} + e$	$Ax + e = b$
dimensions of a square A	$n \times n$	$N \times N$
dimensions of an image X	$\sqrt{n} \times \sqrt{n}$	$n \times n$
Tikhonov solution	$x_{\lambda,L}^{Tik}$	$x_{L,\lambda}$
dimensions of the reg. matrix L	$q \times n$	$M \times N$
TV solution	x_k^{TV}	$x_{\mathrm{TV},k}$
p norm	ℓ_p	ℓ^p
iterative method solution	x_k	$x^{(i)}$
iterative-dependent reg. matrix	W_k	$W^{(i)}$
standard form transformation	$x_{\lambda,L}^{Tik} = L_A^\dagger \bar{x}_{\lambda,L}^{Tik} + x_0$	$x_{L,\lambda} = L_A^\dagger \bar{y}_{L,\lambda} + x_0$ $\bar{x}_{L,\lambda} = L_A^\dagger \bar{y}_{L,\lambda}$

4.2 Published paper

Statement of Authorship

This declaration concerns the article entitled:			
Flexible GMRES for total variation regularization			
Publication status (tick one) **Draft manuscript** ☐ **Submitted** ☐ **In review** ☐ **Accepted** ☐ **Published** X			
Publication details (reference)	S. Gazzola and M. Sabaté Landman. Flexible GMRES for total variation regularization. *Bit Numer Math* 59, 721–746 (2019). https://doi.org/10.1007/s10543-019-00750-x		
Copyright status (tick the appropriate statement) I hold the copyright for this material X Copyright is retained by the publisher, but I have been given permission to replicate the material here ☐			
Candidate's contribution to the paper (provide details, and also indicate as a percentage)	All authors contributed equally to the presentation of the content (50%) The author of the thesis has been the main contributor to the numerical experiments (70%)		
Statement from Candidate	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature.		
Signed	*Malena Sabaté Landman*	**Date**	10th Feb 2021

Last update: Feb 2019

BIT Numerical Mathematics (2019) 59:721–746
https://doi.org/10.1007/s10543-019-00750-x

BIT

Flexible GMRES for total variation regularization

Silvia Gazzola[1] · **Malena Sabaté Landman**[1]

Received: 20 June 2018 / Accepted: 25 March 2019 / Published online: 2 April 2019

Abstract
This paper presents a novel approach to the regularization of linear problems involving total variation (TV) penalization, with a particular emphasis on image deblurring applications. The starting point of the new strategy is an approximation of the non-differentiable TV regularization term by a sequence of quadratic terms, expressed as iteratively reweighted 2-norms of the gradient of the solution. The resulting problem is then reformulated as a Tikhonov regularization problem in standard form, and solved by an efficient Krylov subspace method. Namely, flexible GMRES is considered in order to incorporate new weights into the solution subspace as soon as a new approximate solution is computed. The new method is dubbed TV-FGMRES. Theoretical insight is given, and computational details are carefully unfolded. Numerical experiments and comparisons with other algorithms for TV image deblurring, as well as other algorithms based on Krylov subspace methods, are provided to validate TV-FGMRES.

Keywords TV regularization · Flexible GMRES · Smoothing-norm preconditioning · Image deblurring

Mathematics Subject Classification AMS 65F08 · AMS 65F10 · AMS 65F22

1 Introduction

This paper considers large-scale discrete ill-posed problems of the form

$$b = Ax + e\,, \tag{1.1}$$

where the matrix $A \in \mathbb{R}^{N\times N}$ is ill-conditioned with ill-determined rank (i.e., the singular values of A quickly decay and cluster at zero without an evident gap between two

Communicated by Rosemary Anne Renaut.

Silvia Gazzola
S.Gazzola@bath.ac.uk

[1] Department of Mathematical Sciences, University of Bath, Bath, UK

consecutive ones), and $e \in \mathbb{R}^N$ is unknown Gaussian white noise. Systems like this typically arise when discretizing inverse problems, which are central in many applications (such as astronomical and biomedical imaging, see [11,18] and the references therein). This paper mainly deals with signal deconvolution (deblurring) problems, where $x \in \mathbb{R}^N$ is the unknown sharp signal we wish to recover, and b is the measured (blurred and noisy) signal. In the case of images (i.e., two-dimensional signals) we use the following convention: $X \in \mathbb{R}^{n\times n}$ is the array storing the sharp image, while $x \in \mathbb{R}^N$, $N = n^2$, is the vector obtained by stacking the columns of X. The spatially invariant convolution kernel (blur) is assumed to be known. More specifically, in image deblurring, A is determined starting from a so-called point spread function (PSF, which specifies how the points in the image are distorted by the blur) and the boundary conditions (which specify the behavior of the image outside the recorded values).

Because of the ill-conditioning of A and the presence of noise e, some regularization must be applied to (1.1) in order to compute a meaningful approximation of x. To this end, one may employ the well-known Tikhonov method in general form, which computes a regularized solution

$$x_{L,\lambda} = \arg\min_{x\in\mathbb{R}^N} \|Ax - b\|_2^2 + \lambda\|Lx\|_2^2 . \tag{1.2}$$

Here $L \in \mathbb{R}^{M\times N}$ is the so-called regularization matrix that enforces some smoothing in $x_{L,\lambda}$ (by including the penalization $\|Lx\|_2^2$ in the above objective function), and $\lambda > 0$ is the so-called regularization parameter that specifies the amount of smoothing (by balancing the fit-to-data term $\|Ax - b\|_2^2$ and the regularzation term $\|Lx\|_2^2$). The choice of L and λ is problem-dependent, the former being typically the identity ($L = I$, in which case (1.2) is said to be in standard form), or a rescaled finite difference discretization of a derivative operator. When information about the regularity of x is available, one obtains enhanced reconstructions by considering a suitable $L \neq I$. If the GSVD of the matrix pair (A, L) (or the SVD of the matrix A when $L = I$) can be feasibly computed, then the vector $x_{L,\lambda}$ in (1.2) can be directly expressed as a linear combination of the right (generalized) singular vectors basis. In particular, by using the GSVD, one can notice that additional smoothing is encoded in the basis vectors, and the components of $x_{L,\lambda}$ belonging to the null space of L are unaffected by regularization (see [18, Chapter 8] for more details).

Unfortunately, when considering large-scale problems whose associated coefficient matrix A may not have an exploitable structure or may not be explicitly stored, one cannot assume the GSVD to be available. In this setting iterative regularization methods are the only option, i.e., one can either solve the Tikhonov-regularized problem (1.2) iteratively, or apply an iterative solver to the original system (1.1) and terminate the iterations early (see [4,11,16,18] and the references therein).

This paper considers the last approach, sometimes referred to as "regularizing iterations", and it focuses on the GMRES method [27, Chapter 6] and some variants thereof. GMRES does not require A^T nor matrix-vector products with A^T, and therefore it appears computationally attractive when compared to other regularizing Krylov subspace methods such as LSQR [26]. Although GMRES was proven to be a regu-

larization method in [6], it is well known that it may have a poor performance in some situations, e.g., when dealing with highly non-normal linear systems [21]. It has been shown that, however, this issue can be fixed by using specific preconditioners. For instance, the so-called smoothing-norm preconditioned GMRES method derived in [19] (and here referred to as GMRES(L)) follows from transforming the general Tikhonov problem (1.2) into standard form (i.e., into an equivalent Tikhonov problem with $L = I$), and then applying GMRES to the transformed fit-to-data term. GMRES(L) can be regarded as a right-preconditioned GMRES method that computes an approximate regularized solution as a linear combination of vectors that incorporate the smoothing effect of the regularization matrix L in (1.2). We emphasize that, here and in the following, the term "preconditioner" is used in a somewhat unconventional way. Indeed, the preconditioners used in this paper aim at computing a good regularized solution to problem (1.1) and, from a Bayesian point of view, they may be regarded as "priorconditioners" [5].

Total variation regularization is very popular when dealing with signal deconvolution problems (see [9, Chapter 5] and the references therein), and amounts to computing

$$x_{\mathrm{TV},\lambda} = \arg\min_{x\in\mathbb{R}^N} \|Ax - b\|_2^2 + \lambda \mathrm{TV}(x), \tag{1.3}$$

where $\mathrm{TV}(x)$ denotes the isotropic total variation of the unknown x, which measures the magnitude of the discrete gradient of x in the ℓ^1 norm. The weighted term $\lambda\mathrm{TV}(x)$ in the above Tikhonov-like problem has the effect of producing piecewise-constant reconstructions, as solutions with many steep changes in the gradient are penalized or, equivalently, solutions with a sparse gradient are enforced. In particular, for images, $\lambda\mathrm{TV}(x)$ helps preserving edges.

The convex optimization problem (1.3) is very challenging to solve, both because of its large-scale nature, and because of the presence of the non-differentiable total variation term (so that the efficient iterative techniques used to solve problem (1.2) cannot be straightforwardly adopted in this setting). We also mention in passing that the so-called TV_p penalization term, which evaluates the magnitude of the gradient with respect to some ℓ^p "norm", $0 < p < 1$, can be considered instead of the usual $\mathrm{TV} = \mathrm{TV}_1$, see [7]. TV_p is notably more effective in enforcing sparse gradients (as it better approximates the ℓ^0 quasi-norm), but the resulting Tikhonov-like problem is not convex anymore (and therefore may have multiple local minima). A variety of numerical approaches for the solution of (1.3) have already been proposed: some of them are based on fixed-point iterations, smooth approximations of $\mathrm{TV}(x)$, fast gradient-based iterations, and Bregman-distance methods; see [3,8,25,29], to cite only a few.

This paper is concerned with strategies that stem from the local approximation of (1.3) by a sequence of quadratic problems of the form (1.2), and that exploit Krylov subspace methods to compute solutions thereof. To the best of our knowledge, this idea was first proposed for total variation regularization in [30], where the authors derive the so-called iteratively reweighted norm (IRN) method consisting of the solution of a sequence of penalized weighted least-squares problems with diagonal weighting matrices incorporated into the regularization term and dependent on the previous

approximate solution (so that they are updated from one least-squares problem to the next one). For large-scale unstructured problems, this method intrinsically relies on an inner-outer iteration scheme. In the following we use the acronym IRN to indicate a broad class of methods that can be recast in this framework.

Although the IRN method [30] is theoretically well-justified and experimentally effective, it has a couple of drawbacks. Firstly, conjugate gradient is repeatedly applied from scratch to the normal equations associated to each penalized least-squares problem of the form (1.2) in the sequence: this may result in an overall large number of iterations. Secondly, the regularization parameter λ should be chosen (and fixed) in advance. The so-called modified LSQR (MLSQR) method [1] partially remedies both these shortcomings. Although the starting point of MLSQR is still an IRN approach [30], each Tikhonov-regularized problem in the sequence of least-squares problems is transformed into standard form: in this way the matrix A is now right preconditioned and a preconditioned LSQR method can be applied. This approach typically results in a smaller number of iterations with respect to IRN [30]; moreover, different values of the regularization parameter can be easily considered. On the downside, LSQR is still applied sequentially to each IRN least-squares problem, and a new approximation subspace for the LSQR solution is computed from scratch. The so-called GKSpq method [23] leverages generalized Krylov subspaces (GKS), i.e., approximation subspaces where the updated weights and adaptive regularization parameters can be easily incorporated as soon as they become available. In other words, only one approximation subspace is generated when running the GKSpq method for the IRN least-squares problems associated to (1.3), and the approximate solutions are obtained by orthogonal projections onto GKS of increasing dimension. In this way, GKSpq avoids inner-outer iterations and is very efficient when compared to IRN and MLSQR.

All the methods surveyed so far implicitly consider the normal equations associated to least-squares approximations of problem (1.3). As already remarked, approaches based on GMRES applied directly to the fit-to-data term in (1.2) may be more beneficial in some situations, as the computational overload of dealing with A^T can be avoided. The restarted generalized Arnoldi–Tikhonov (ReSt-GAT) method [15] is arguably the only approach that generates a GMRES-like approximation subspace for the solution of each least-squares problem associated to the IRN strategy. However ReSt-GAT has two shortcomings: it is based on an inner-outer iteration scheme (though approximations recovered during an iteration cycle are carried over to the next one by performing convenient warm restarts) and the TV penalization does not directly affect the approximation subspace of ReSt-GAT (failing to properly enhance piecewise constant reconstructions).

The goal of this paper is to propose a novel strategy that employs GMRES for the solution of Tikhonov-regularized problems associated to the IRN approach to (1.3). In particular, a flexible instance of a GMRES(L)-like method is used to solve preconditioned versions of system (1.1), which are obtained by considering quadratic approximations to problem (1.3), performing transformations into standard form, and applying GMRES to the resulting fit-to-data term. In this way, the effect of the total variation regularization term (defined with respect to iteratively updated weights and a discrete gradient operator) is incorporated into the solution subspace, which is affected by both the null space of the regularization matrix and the adaptive weights. As the

weights are updated as soon as a new approximate solution becomes available, i.e., immediately after a new GMRES iteration is computed, the flexible GMRES (FGMRES) method (see [27, Chapter 9]) is employed to handle variable preconditioning along the iterations. The resulting regularization method is dubbed Total-Variation-FGMRES (TV-FGMRES). We emphasize that the TV-FGMRES method is inherently parameter-free, as only one stopping criterion should be set to suitably terminate the iterations (while, for all the other solvers for problem (1.3) listed so far, one has to choose both the parameter λ and the number of iterations). Moreover, the new approach is different from the ReSt-GAT one [15] for two reasons: firstly, the standard GMRES approximation subspaces are modified and, secondly, regularizing iterations are employed rather than solving a sequence of Tikhonov problems (1.2); also, this approach is somewhat analogous to the GKSpq [23] one, but the two methods differ in the computation of the approximation subspaces (recall that the GKSpq ones involve both A^T and λ).

This paper is organized as follows. Section 2 covers some background material, including the definition of the weighting matrices for the approximation of the total variation regularization term in an IRN fashion, and a well-known procedure for transforming problem (1.2) into standard form. Section 3 describes the new TV-FGMRES method. Section 4 dwells on implementation details. Section 5 contains numerical experiments performed on three different image deblurring test problems. Section 6 presents some concluding remarks and possible future works.

2 IRN, weights, and standard form transformation

The main idea underlying the IRN approaches for the solution of TV-regularized problems is to approximate the minimizer of (1.3) by solving a sequence of regularized problems with rescaled penalization terms expressed as reweighted ℓ^2 norms, whose weights are iteratively updated using a previous approximation of the solution, i.e.,

$$x_{\mathrm{TV},\lambda} \simeq x_{L,\lambda} = \arg\min_{x\in\mathbb{R}^N} \|Ax-b\|_2^2+\lambda\|Lx\|_2^2\,, \quad \text{with} \quad L = WD \in \mathbb{R}^{M\times N}\,, \tag{2.1}$$

where $W = W^{(i)} = W(Dx^{(i-1)})$ denotes a diagonal weighting matrix defined with respect to an available approximation $x^{(i-1)}$ of $x_{\mathrm{TV},\lambda}$ (so that also L depends on $x^{(i-1)}$), and D denotes a scaled finite difference matrix discretizing a derivative operator of the first order. More precisely, (2.1) is obtained by locally approximating the total variation functional in (1.3) by the quadratic functional

$$\frac{1}{2}\left(||W(Dx^{(i-1)})Dx||_2^2 + \mathrm{TV}(x^{(i-1)})\right)\,, \tag{2.2}$$

where the constant second term is dropped and the rescaling factor is incorporated into λ in (2.1). The weights $W(Dx^{(i-1)})$ are determined in such a way that (2.2) is tangent to $\mathrm{TV}(x)$ at $x = x^{(i-1)}$, and it is an upper bound for $\mathrm{TV}(x)$ elsewhere, see [30]. The weights are derived as follows, distinguishing between:

– The *one-dimensional* (*1d*) case. In a discrete setting with $v \in \mathbb{R}^N$, $\mathrm{TV}(v) = \|D_{1d} v\|_1$, where

$$D_{1d} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(N-1)\times N}. \tag{2.3}$$

The weighting matrix

$$W_{1d} = W_{1d}(D_{1d} v) = \mathrm{diag}\left(|D_{1d} v|^{-1/2}\right) \in \mathbb{R}^{(N-1)\times(N-1)}, \tag{2.4}$$

where both modulus and exponentiation are considered component-wise, is used in practice to approximate the 1-norm. Indeed, for a given v, one can easily see that

$$\|W_{1d} D_{1d} v\|_2^2 = \sum_{k=1}^{N-1} |[D_{1d} v]_k|^{-1} [D_{1d} v]_k^2 = \sum_{k=1}^{N-1} |[D_{1d} v]_k| = \|D_{1d} v\|_1 = \mathrm{TV}(v),$$

where $[w]_k$ denotes the kth entry of a vector w.

– The *two-dimensional* (*2d*) case. In a discrete setting,

$$\mathrm{TV}(v) = \left\| \left((D^{\mathrm{h}} v)^2 + (D^{\mathrm{v}} v)^2 \right)^{1/2} \right\|_1,$$

where, if $v \in \mathbb{R}^N$ is obtained by stacking the columns of a 2d array $V \in \mathbb{R}^{n\times n}$ with $N = n^2$, the discrete first derivatives in the horizontal and vertical directions are given by

$$D^{\mathrm{h}} = (D_{1d} \otimes I) \in \mathbb{R}^{n(n-1)\times n^2}, \qquad D^{\mathrm{v}} = (I \otimes D_{1d}) \in \mathbb{R}^{n(n-1)\times n^2},$$

respectively. Here $D_{1d} \in \mathbb{R}^{(n-1)\times n}$ is the 1d first derivative matrix (2.3) of appropriate size, and I is the identity matrix of size n, so that 2d discrete operators are defined in terms of the corresponding 1d ones (note that here both D_{1d} and I have n columns, i.e., the size of the 2d array V). Deriving an expression for the weights in the discrete 2d setting is less straightforward. Following [30], for a given v, and letting $\widetilde{N} = n(n-1)$, one takes

$$\begin{aligned} D_{2d} &= \begin{bmatrix} D^{\mathrm{h}} \\ D^{\mathrm{v}} \end{bmatrix} \in \mathbb{R}^{2\widetilde{N}\times N}, \\ \widetilde{W}_{2d} &= \widetilde{W}_{2d}(D_{2d} v) = \mathrm{diag}\left(\left((D^{\mathrm{h}} v)^2 + (D^{\mathrm{v}} v)^2 \right)^{-1/4} \right) \in \mathbb{R}^{\widetilde{N}\times\widetilde{N}}, \\ W_{2d} &= W_{2d}(D_{2d} v) = \begin{bmatrix} \widetilde{W}_{2d} & 0 \\ 0 & \widetilde{W}_{2d} \end{bmatrix} \in \mathbb{R}^{2\widetilde{N}\times 2\widetilde{N}}. \end{aligned} \tag{2.5}$$

The expression of the weights in (2.4) and (2.5) generalizes to the TV_p functional, $0 < p < 1$, defined as

$$\mathrm{TV}_p(v) = \|D_{1\mathrm{d}} v\|_p^p \quad \text{and} \quad \mathrm{TV}_p(v) = \left\| \left((D^{\mathrm{h}} v)^2 + (D^{\mathrm{v}} v)^2 \right)^{1/2} \right\|_p^p ,$$

in the 1d and 2d cases, respectively. Indeed, given v, it suffices to take the weights

$$\begin{aligned} W_{p,1\mathrm{d}}(D_{1\mathrm{d}} v) &= \mathrm{diag}\left(|D_{1\mathrm{d}} v|^{(p-2)/2} \right) \quad \text{and} \\ \widetilde{W}_{p,2\mathrm{d}}(D_{2\mathrm{d}} v) &= \mathrm{diag}\left(\left((D^{\mathrm{h}} v)^2 + (D^{\mathrm{v}} v)^2 \right)^{(p-2)/4} \right) , \end{aligned}$$

for the 1d and 2d cases, respectively, and then proceed as in the TV case illustrated above.

It is important to stress that division by zero may occur when computing the weights associated to the TV and TV_p functionals (this is the case when a component of the gradient magnitude is zero, which should not be regarded and a rare occurrence as (1.3) enforces sparsity in the gradient of the solution). To avoid this, one should set some safety thresholds $\tau_1 > \tau_2 > 0$, define

$$f_\tau([w]_k) = \begin{cases} |[w]_k| & \text{if } |[w]_k| > \tau_1 \\ \tau_2 & \text{otherwise} \end{cases} \qquad \text{for each component } [w]_k \text{ of a vector } w, \tag{2.6}$$

and then consider as weights the diagonal matrices $W_{1\mathrm{d}}(f_\tau(D_{1\mathrm{d}} v))$ and $W_{2\mathrm{d}}(f_\tau(D_{2\mathrm{d}} v))$ for TV, $W_{p,1\mathrm{d}}(f_\tau(D_{1\mathrm{d}} v))$ and $W_{p,2\mathrm{d}}(f_\tau(D_{2\mathrm{d}} v))$ for TV_p.

In the following, when the distinction between the 1d and the 2d cases can be waived, we will use the simpler notations W for the $M \times M$ diagonal weighting matrix, and D for the $M \times N$ first derivative matrix (with $M = N - 1$ in the 1d case, and $M = 2\widetilde{N}$ in the 2d case).

We conclude this section by recalling a well-known strategy to transform a generalized Tikhonov-regularized problem (1.2) with a given regularization matrix $L \in \mathbb{R}^{M \times N}$ into standard form, with $\mathrm{rank}(L) = M < N$ and $\mathscr{N}(A) \cap \mathscr{N}(L) = \{\mathbf{0}\}$, i.e., the null spaces of A and L trivially intersect (see [12] for more details). Under these assumptions, the solution of (1.2) can be equivalently expressed as

$$x_{L,\lambda} = L_A^\dagger \bar{y}_{L,\lambda} + x_0 = \bar{x}_{L,\lambda} + x_0 , \quad \text{where} \quad \begin{aligned} \bar{y}_{L,\lambda} &= \arg\min_{\bar{y} \in \mathbb{R}^M} \|\bar{A}\bar{y} - \bar{b}\|_2^2 + \lambda \|\bar{y}\|_2^2 , \\ \bar{A} &= A L_A^\dagger , \\ \bar{b} &= b - A x_0 . \end{aligned} \tag{2.7}$$

Here $L_A^\dagger$ is the A-weighted pseudoinverse of L, defined as

$$L_A^\dagger = [I - (A(I - L^\dagger L))^\dagger A] L^\dagger \in \mathbb{R}^{N \times M},$$

where $L^\dagger$ denotes the Moore–Penrose pseudoinverse of L, $\bar{x}_L$ is the component of the solution x_L in $\mathscr{R}(L_A^\dagger)$ (i.e., the range of $L_A^\dagger$), and x_0 is the component of the solution

x_L in $\mathscr{N}(L)$ (i.e., $x_0 = (A(I - L^\dagger L))^\dagger b$). We remark that, when a given matrix $L \in \mathbb{R}^{M\times N}$ has full rank but is such that $M > N$, a reduced QR factorization of L, $L = QR$, can be performed, and the above derivations can be applied to $R \in \mathbb{R}^{N\times N}$ (instead of L). In the following we will use this procedure to transform (2.1), for a given weighting matrix $W = W^{(i)} = W(Dx^{(i-1)})$, into standard form.

3 TV-preconditioned flexible GMRES

We begin this section by defining a TV-preconditioned version of the GMRES(L) method [19], motivated by the generalized Tikhonov problem (2.1) appearing within the IRN framework and assuming that the matrix $W = W^{(i)} = W(Dx^{(i-1)})$ is fixed. To achieve this, we first transform problem (2.1) into standard form, following the procedure outlined in (2.7). We then describe how to apply GMRES to a system linked to the fit-to-data term in (2.7) or, in other words, we consider $\lambda = 0$ in (2.7). Indeed, we would like to apply GMRES (as a regularizing iterative method) to the system

$$A(L_A^\dagger \bar{y}_L + x_0) = b\,, \tag{3.1}$$

where $L_A^\dagger$ can be regarded as a preconditioner accounting for a total variation regularization term, x_0 is defined as in (2.7), and $\bar{y}_L$=$\bar{y}_{L,0}$ (note that, to keep the notations simple, in the following we will use $\bar{y}_L$=$\bar{y}_{L,0}$, $\bar{x}_L$=$\bar{x}_{L,0}$, and x_L=$x_{L,0}$).

Applying GMRES to (3.1) is not straightforward, because the coefficient matrix $AL_A^\dagger \in \mathbb{R}^{N\times M}$ is rectangular. To overcome this obstacle we closely follow the approach proposed in [19]. Let K be a matrix with full column-rank whose columns span $\mathscr{N}(L)$: when L is as in (2.1), since W is nonsingular, this means that

$$\mathscr{R}(K) = \mathscr{N}(L) = \mathscr{N}(D) = \text{span}\{\mathbf{1}\}$$

for both the 1d and the 2d cases, where $\mathbf{1} = [1, \ldots, 1]^T \in \mathbb{R}^N$. Thanks to this remark, the expression for x_L in (2.7) (with $\lambda = 0$) can be further detailed as

$$x_L = \bar{x}_L + x_0 = L_A^\dagger \bar{y}_L + x_0 = L_A^\dagger \bar{y}_L + Kt_0\,, \tag{3.2}$$

where the scalar $t_0 \in \mathbb{R}$ is uniquely determined by computing

$$t_0 = (AK)^\dagger b = \arg\min_{t\in\mathbb{R}} \|(AK)t - b\|_2\,. \tag{3.3}$$

We note that decomposition (3.2) is uniquely determined if both L and K have full rank. By plugging expression (3.2) into (3.1), we get

$$A\begin{bmatrix} L_A^\dagger, & K \end{bmatrix}\begin{bmatrix} \bar{y}_L \\ t_0 \end{bmatrix} = b\,, \tag{3.4}$$

which, once premultiplied by $[D^\dagger,\ K]^T$, gives the 2×2 block system

$$\begin{bmatrix} (D^\dagger)^T A L_A^\dagger & (D^\dagger)^T A K \\ K^T A L_A^\dagger & K^T A K \end{bmatrix} \begin{bmatrix} \bar{y}_L \\ t_0 \end{bmatrix} = \begin{bmatrix} (D^\dagger)^T b \\ K^T b \end{bmatrix}.$$

We can easily eliminate t_0 from this system by inverting the 1×1 $(2, 2)$ block, so obtaining the Schur complement system,

$$(D^\dagger)^T P A L_A^\dagger \bar{y}_L = (D^\dagger)^T P b\,, \tag{3.5}$$

or, using once more the relations in (3.2),

$$(D^\dagger)^T P A \bar{x}_L = (D^\dagger)^T P b\,, \tag{3.6}$$

where

$$P = I - AK(K^T AK)^{-1} K^T \in \mathbb{R}^{N \times N} \tag{3.7}$$

is the oblique projector onto the orthogonal complement of $\mathscr{R}(K)$ along $\mathscr{R}(AK)$. The coefficient matrix in system (3.5) has size $M \times M$, while the one in system (3.6) has size $M \times N$. We emphasize that this Schur complement system is different from the one derived in [19], where both sides of (3.4) are premultiplied by the matrix $[L_A^\dagger,\ K]^T$. In our setting, since the matrix L contains weights W that should be suitably updated (as explained in the remaining part of this section), we conveniently discard the contribution of W in the left preconditioner in (3.5). In the following, to keep the formulas light, we often use the notations

$$\widehat{A} = (D^\dagger)^T P A \in \mathbb{R}^{M \times N}\,, \quad \widehat{b} = (D^\dagger)^T P b \in \mathbb{R}^M\,, \tag{3.8}$$

so that systems (3.5) and (3.6) can be even more compactly written as $\widehat{A} L_A^\dagger \bar{y}_L = \widehat{b}$ and $\widehat{A} \bar{x}_L = \widehat{b}$, respectively. The mth iteration of the GMRES method applied to compute x_L as in (3.2) produces an approximation $x_{L,m}$ thereof, such that

$$x_{L,m} \in L_A^\dagger \mathscr{K}_m((D^\dagger)^T P A L_A^\dagger, (D^\dagger)^T P b) + x_0 = \mathscr{K}_m(L_A^\dagger (D^\dagger)^T P A, L_A^\dagger (D^\dagger)^T P b) + x_0\,.$$

We stress again that here x_0 is the component of x_L in $\mathscr{N}(L)$, and the weighting matrix W (implicit in L) is fixed. Also, referring to the original notations in (2.7), $x_{L,m} = x_{L,0,m}$ (i.e., $\lambda = 0$).

Dropping the assumption that the weights are fixed, according to the IRN framework we should keep updating L in (2.1) with new approximations of x_L. If this happens as soon as a new approximation $x_{L,m-1}$ is computed by the GMRES method applied to (3.5), i.e., if $W = W(Dx_{L,m-1})$, then we should incorporate an iteration-dependent A-weighted pseudo-inverse (denoted by $(L^{(m)})_A^\dagger$) within the GMRES scheme to solve (3.5), and adopt a flexible version of the Arnoldi algorithm [27, Chapter 9] to handle

variable preconditioning: this leads to the TV-FGMRES method. The mth iteration of the flexible Arnoldi algorithm updates a decomposition of the form

$$\widehat{A}Z_m = V_{m+1}H_m\,, \quad \text{where} \quad Z_m \in \mathbb{R}^{N\times m},\ V_{m+1} \in \mathbb{R}^{M\times(m+1)},\ H_m \in \mathbb{R}^{(m+1)\times m}. \tag{3.9}$$

More specifically, in the above decomposition: H_m is an upper Hessenberg matrix; V_{m+1} has orthonormal columns v_i, $i = 1,\dots,m+1$, with $v_1 = \widehat{b}/\|\widehat{b}\|_2$; Z_m has columns $z_i = (L^{(i)})_A^\dagger v_i$, $i = 1,\dots,m$. Since the columns of Z_m already include the contribution of the variable preconditioners $(L^{(i)})_A^\dagger$, for $i = 1,\dots,m$, they form a basis for the vector $\bar{x}_L$ in (3.2). Therefore, at the mth step of TV-FGMRES, $\bar{x}_L$ is approximated by the following vector

$$\bar{x}_{L,m} = Z_m s_m\,, \quad \text{where} \quad s_m = \arg\min_{s\in\mathbb{R}^m} \|H_m s - \|\widehat{b}\|_2 e_1\|_2\,, \tag{3.10}$$

and $e_1 \in \mathbb{R}^{m+1}$ is the first canonical basis vector of $\mathbb{R}^{m+1}$. Due to decomposition (3.9) and the properties of the matrices appearing therein,

$$\begin{aligned}\min_{\bar{x}_L\in\mathscr{R}(Z_m)} \|\widehat{A}\bar{x}_L - \widehat{b}\|_2 &= \min_{s\in\mathbb{R}^m} \|\widehat{A}Z_m s - \widehat{b}\|_2 = \min_{s\in\mathbb{R}^m} \|V_{m+1}(H_m s - \|\widehat{b}\|_2 e_1)\|_2 \\ &= \min_{s\in\mathbb{R}^m} \|H_m s - \|\widehat{b}\|_2 e_1\|_2\,,\end{aligned}$$

i.e., the approximate solution $\bar{x}_{L,m}$ obtained at the mth iteration of TV-FGMRES minimizes the residual norm of (3.6) over all the vectors in

$$\mathscr{R}(Z_m) = \text{span}\{(L^{(1)})_A^\dagger v_1, (L^{(2)})_A^\dagger v_2, (L^{(3)})_A^\dagger v_3, \dots\}\,. \tag{3.11}$$

The approximation subspace $\mathscr{R}(Z_m)$ can be regarded as a preconditioned Krylov subspace, where the preconditioner is implicitly defined by successive applications of the matrices $(L^{(i)})_A^\dagger$ to the linearly independent vectors v_i; see [24].

The main steps of the TV-FGMRES method are summarized in Algorithm 1 where, notation-wise, $[B]_{j,i}$ denotes the (j,i)th entry of a matrix B.

If $(L^{(i)})_A^\dagger = L_A^\dagger$ is fixed, then

$$\begin{aligned}\mathscr{R}(Z_m) &= \mathscr{K}_m(L_A^\dagger (D^\dagger)^T PA, L_A^\dagger (D^\dagger)^T Pb) \\ &= L_A^\dagger \mathscr{K}_m((D^\dagger)^T PAL_A^\dagger, (D^\dagger)^T Pb) = L_A^\dagger \mathscr{R}(V_m),\end{aligned}$$

where V_m is the orthonormal basis generated by the right-preconditioned Arnoldi algorithm, which can be expressed by a decomposition formally similar to (3.9), but where $Z_m = L_A^\dagger V_m$. Recalling previous naming conventions, $\mathscr{R}(Z_m) = L_A^\dagger \mathscr{R}(V_m)$ is the approximation subspace associated to the GMRES(WD) method.

We conclude this section by proposing a simple numerical experiment to illustrate the benefit of considering flexibility, i.e., TV-FMGRES, over non-flexible versions of GMRES. We consider a signal deblurring test problem, where a piecewise constant sharp signal of length $N = 256$ is convolved with a Gaussian blurring kernel, and

Algorithm 1 The TV-FGMRES method.

Input: $\widehat{A}, \widehat{b}$ as in (3.8)
Take $W^{(1)} = I$ and $v_1 = \widehat{b}/||\widehat{b}||_2$
Compute x_0 in (3.2) by (3.3)
For $i = 1, \ldots$ until a stopping criterion is satisfied

1. Compute $z_i = (L^{(i)})_A^\dagger v_i = (W^{(i)} D)_A^\dagger v_i$
2. Compute $w = \widehat{A} z_i$
3. For $j = 1, \ldots, i$, compute $[H]_{j,i} = w^T v_i$ and set $w = w - [H]_{j,i} v_j$
4. Compute $[H]_{i+1,i} = ||w||_2$ and, if $[H]_{i+1,i} \neq 0$, take $v_{i+1} = w/[H]_{i+1,i}$
5. Compute the solution s_i in (3.10) so that $\bar{x}_{L,i} = Z_i s_i$
6. Update the weights $W^{(i+1)}$

end
Take $x_L = \bar{x}_{L,i} + x_0$

Gaussian white noise of level 10^{-2} is added, so to obtain a corrupted version of it (as reported in Fig. 1a). We consider the following iterative approaches to recover the exact signal: GMRES, GMRES(D_{1d}), and TV-GMRES (with thresholds $\tau_1 = 10^{-4}$, $\tau_2 = 10^{-12}$); for the sake of comparison, we also include GMRES($W^{\text{ex}} D_{1d}$), where $W^{\text{ex}} = W(Dx^{\text{ex}})$ are the (optimal) weights computed with respect to the exact solution x^{ex} of the noise-free problem (i.e., (1.1) with $e = 0$). In Fig. 1b the best reconstructions obtained by GMRES, GMRES(D_{1d}), and TV-FGMRES are reported: it is evident that the latter is the most capable method in reproducing the piecewise constant features of the original signal, as the GMRES and the GMRES(D_{1d}) reconstructions are affected by many spurious oscillations. Indeed, the combination of flexibility and appropriate adaptive weightings allows to generate basis vectors for TV-FGMRES that are the closest to the optimal ones (see Fig. 1e, f). The basis vectors for GMRES immediately exhibit an extremely oscillatory behavior (see Fig. 1c), while the basis vectors for GMRES(D_{1d}) are greatly smoothed, but fail to reproduce the jumps characterizing the exact signal (see Fig. 1d). We can experimentally conclude that one of the reasons underlying the success of TV-FGMRES is the iteration-dependent preconditioning that enforces piecewise-constant features of the solution into the TV-FGMRES basis vectors.

4 Implementation strategies

To devise efficient implementations of the TV-FGMRES method applied to the system (3.5) or (3.6), a number of properties of the involved matrices should be taken into account. In this section we will often use MATLAB-like notations: for instance, we will use a dot to denote a component-wise operation, a colon to access elements in a range of rows or columns of an array, and diag(·) to denote a vector of diagonal entries. We will extensively invoke and generalize some of the propositions derived in [19]. We start by proving the following result for system (3.5) (analogous to Theorem 5.1 in [19]).

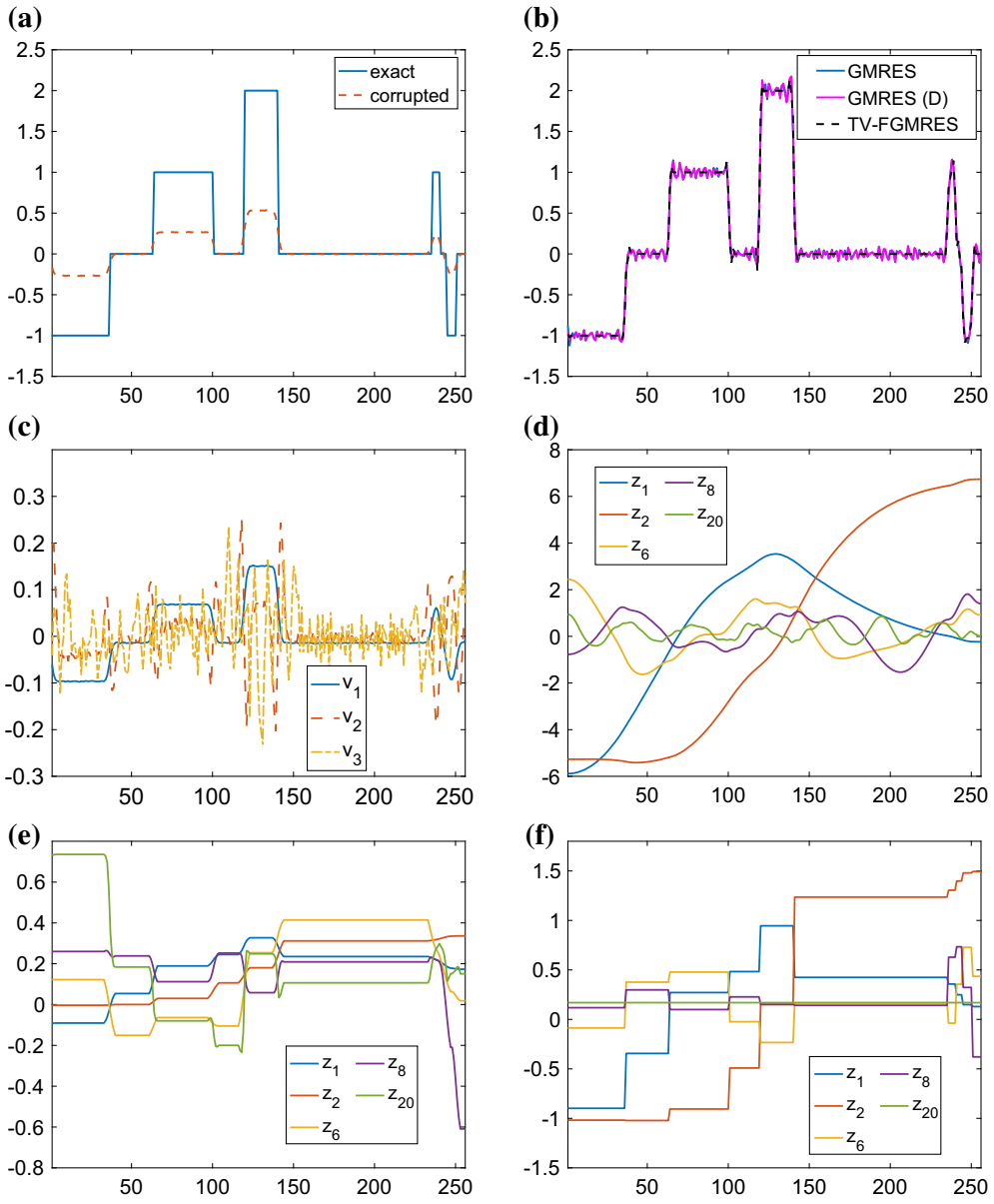


Fig. 1 Signal deblurring problem. **a** 1d piecewise constant exact signal x^{ex} and its corrupted version. **b** Best reconstructions obtained by the GMRES, GMRES(D_{1d}), and TV-FGMRES methods. **c** Basis vectors v_1, v_2, v_3 for GMRES. **d** Basis vectors z_1, z_2, z_6, z_8, z_{20} for GMRES(D_{1d}). **e** Basis vectors z_1, z_2, z_6, z_8, z_{20} for TV-FGMRES. **f** Basis vectors z_1, z_2, z_6, z_8, z_{20} for GMRES($W^{\text{ex}} D_{\text{1d}}$)

Theorem 4.1 *If $\mathscr{R}(L^T)$ and $\mathscr{R}(AK)$ are complementary subspaces, the Schur complement system (3.5) is equivalent to*

$$(D^\dagger)^T P A L^\dagger y = (D^\dagger)^T P b\,, \tag{4.1}$$

where P is given by (3.7).

Proof We start by noting that $L_A^\dagger = EL^\dagger$, where

$$E = I - (A(I - L^\dagger L))^\dagger A = I - (AKK^\dagger)^\dagger A = I - K(AK)^\dagger A\,. \tag{4.2}$$

Then

$$\begin{aligned}(D^\dagger)^T PAEL^\dagger &= (D^\dagger)^T PA(I - K(AK)^\dagger A)L^\dagger \\ &= (D^\dagger)^T PAL^\dagger - (D^\dagger)^T PAK(AK)^\dagger AL^\dagger\,,\end{aligned}$$

where the second term in the above sum is

$$\begin{aligned}(D^\dagger)^T PAK(AK)^\dagger AL^\dagger &= (D^\dagger)^T (I - AK(K^T AK)^{-1}K^T)AK(AK)^\dagger AL^\dagger \\ &= (D^\dagger)^T AK(AK)^\dagger AL^\dagger \\ &\quad - (D^\dagger)^T AK(K^T AK)^{-1}(K^T AK)(AK)^\dagger AL^\dagger = 0.\end{aligned}$$

Therefore, (3.5) reduces to (4.1). □

Note that, although this theorem is stated for system (3.5), the same relations can be exploited when solving system (3.6), since the matrix-vector products $((D^\dagger)^T PA)(L_A^\dagger v) = ((D^\dagger)^T PA)(L^\dagger v)$ should be computed (see also lines 1, 2 of Algorithm 1). The above theorem is important from a computational point of view since, by avoiding multiplication by E, an additional matrix-vector product with A can be avoided at each iteration of TV-FGMRES. However, in a flexible framework, we may still need the solution $\bar{x}_{L,i}$ (which is implicitly expressed through a matrix-vector product with $L_A^\dagger$) to update the weights $W^{(i+1)}$ at each iteration (see line 5 of Algorithm 1). This is notably not the case for TV-FGMRES, as the weights are expressed with respect to the gradient of $\bar{x}_{L,i}$, and

$$D\bar{x}_{L,i} = DL_A^\dagger \bar{y}_{L,i} = DL^\dagger \bar{y}_{L,i} - DK((AK)^\dagger A\bar{y}_{L,i}) = DL^\dagger \bar{y}_{L,i}\,,$$

where we have used the fact that $\mathscr{R}(K) = \mathscr{N}(L) = \mathscr{N}(D)$. Note also that, to keep the notations light, we have compactly denoted by $L_A^\dagger \bar{y}_{L,i}$ a vector belonging to the space (3.11). Finally, although the matrix P in (3.7) is defined in terms of A, matrix-vector products with A can be smartly avoided when computing matrix-vector products with P, by observing that

$$\begin{aligned}Pv &= v - AK(K^T AK)^{-1}K^T v = v - Q_0 R_0 (K^T Q_0 R_0)^{-1} K^T v \\ &= v - Q_0 (K^T Q_0)^{-1} K^T v\,,\end{aligned}$$

where $AK = Q_0 R_0$ is the reduced QR factorization of $AK \in \mathbb{R}^N$, i.e., $Q_0 \in \mathbb{R}^N$ is the normalization of AK. Matrix-vector products with P have therefore an $O(N)$ cost.

As a consequence, under the assumptions of Theorem 4.1, the computational cost per iteration of TV-FGMRES is dominated by one matrix-vector product with A (similarly to GMRES applied to (1.1)), plus one matrix-vector product with $(D^\dagger)^T$ and

$L^\dagger$. Efficient approaches to compute the latter will be explored in the next subsections. We conclude by remarking that, when considering image deblurring problems with spatially invariant blurs and periodic boundary conditions [20], the normalization condition $A\mathbf{1} = \mathbf{1}$ is satisfied by the blurring matrix A (this is basically a conservation of light condition for the blurred image). In this setting, $\mathscr{R}(L^T)$ and $\mathscr{R}(AK)$ are indeed complementary subspaces, since

$$\begin{aligned}\mathscr{R}(L^T) + \mathscr{R}(AK) = \mathscr{R}(L^T) + A\mathscr{R}(K) = \mathscr{R}(L^T) \\ + A\mathscr{N}(L) = \mathscr{R}(L^T) + \mathscr{N}(L) = \mathbb{R}^N,\end{aligned}$$

and $\mathscr{R}(L^T) \cap \mathscr{N}(L) = \{\mathbf{0}\}$ thanks to the fundamental theorem of linear algebra.

4.1 Computations of matrix-vector products with D and $(D^\dagger)^T$

These are required when computing the weights (defined in (2.4), (2.5)), and when computing matrix-vector products with $\widehat{A}$ (defined in (3.8)). We focus on the 2d case, as special strategies should be used to handle large-scale quantities. Concerning the first task, given a vector $v \in \mathbb{R}^N$, we can exploit the special structure of D_{2d} and the Kronecker product properties, so that

$$\begin{aligned}D_{2d}v = \begin{bmatrix} D^{\mathrm{h}} \\ D^{\mathrm{v}} \end{bmatrix} v = \begin{bmatrix} (D_{1d} \otimes I)v \\ (I \otimes D_{1d})v \end{bmatrix} = \begin{bmatrix} V D_{1d}^T \\ D_{1d} V \end{bmatrix} \\ = \begin{bmatrix} V(:, 1:n-1) & - & V(:, 2:n) \\ V(1:n-1, :) & - & V(2:n, :) \end{bmatrix},\end{aligned} \tag{4.3}$$

where $v \in \mathbb{R}^N$ is obtained by stacking the columns of $V \in \mathbb{R}^{n\times n}$, $N = n^2$, and where columns and rows differences of V are computed. Matrix-vector products with D_{2d} have an $O(n(n-1)) = O(N)$ cost.

Concerning the computation of matrix-vector products with $(D^\dagger)^T$, let us first assume that the singular value decomposition of $D_{1d} = U_{1d}\Sigma_{1d}V_{1d}^T$ can be computed. Then, by using some Kronecker product properties, we can decompose D_{2d} in (2.5) as follows

$$\begin{aligned}D_{2d} = \begin{bmatrix} D_{1d} \otimes I \\ I \otimes D_{1d} \end{bmatrix} = \begin{bmatrix} U_{1d}\Sigma_{1d}V_{1d}^T \otimes V_{1d}V_{1d}^T \\ V_{1d}V_{1d}^T \otimes U_{1d}\Sigma_{1d}V_{1d}^T \end{bmatrix} \\ = \begin{bmatrix} U_{1d} \otimes V_{1d} & 0 \\ 0 & V_{1d} \otimes U_{1d} \end{bmatrix} \widetilde{\Sigma} \left[V_{1d} \otimes V_{1d} \right]^T, \quad \text{where} \quad \widetilde{\Sigma} = \begin{bmatrix} \Sigma_{1d} \otimes I \\ I \otimes \Sigma_{1d} \end{bmatrix}.\end{aligned} \tag{4.4}$$

The matrix $\widetilde{\Sigma}$ has a very sparse structure, where the only nonzero entries are

$$[\widetilde{\Sigma}]_{j,j} \text{ if } j \le n(n-1), \quad \text{and} \quad [\widetilde{\Sigma}]_{n(n-1)+j-\left\lfloor \frac{j}{n} \right\rfloor, j} \text{ if } j < n^2 \text{ and } j \neq 0 \mod n\,.$$

A more convenient decomposition can be devised by applying a set of Givens rotations to the matrix $\widetilde{\Sigma}$, so that the QR decomposition

$$\widetilde{Q}\widetilde{D} = \widetilde{\Sigma} \tag{4.5}$$

is implicitly obtained, where $\widetilde{Q} \in \mathbb{R}^{2\widetilde{N}\times 2\widetilde{N}}$ is an orthogonal matrix and $\widetilde{D} \in \mathbb{R}^{2\widetilde{N}\times N}$ is a nonnegative diagonal matrix of rank $N-1$. By plugging (4.5) into (4.4), and by using standard properties of the pseudoinverse, we get

$$(D_{2\mathrm{d}}^{\dagger})^T = \begin{bmatrix} U_{1\mathrm{d}} \otimes V_{1\mathrm{d}} & 0 \\ 0 & V_{1\mathrm{d}} \otimes U_{1\mathrm{d}} \end{bmatrix} \widetilde{Q}(\widetilde{D}^{\dagger})^T \left[V_{1\mathrm{d}} \otimes V_{1\mathrm{d}} \right]^T . \tag{4.6}$$

Computing matrix-vector products with $(D_{2\mathrm{d}}^{\dagger})^T$ has an overall $O(N^{3/2})$ cost, provided that vectors of length N are conveniently reshaped as $n \times n$ 2D arrays ($N = n^2$) to exploit the Kronecker product properties.

4.2 Computation of matrix-vector products with $L^{\dagger}$ and $L_A^{\dagger}$

Since the A-weighted pseudoinverse can be expressed as $L_A^{\dagger} = EL^{\dagger}$, see (4.2), and since $K \in \mathbb{R}^N$ for TV-FGMRES, the computational burden of matrix-vector products with $L_A^{\dagger}$ mainly lays in the computation of matrix-vector products with A and $L^{\dagger}$. Concerning the latter, we remark that in the 1d case one simply has $(W_{1\mathrm{d}}D_{1\mathrm{d}})^{\dagger} = D_{1\mathrm{d}}^{\dagger}W_{1\mathrm{d}}^{-1}$ directly from the definition of Moore–Penrose pseudoinverse of matrices with linearly independent rows. Unfortunately, in the 2d case, the matrix

$$\widetilde{L}^{\dagger} = D_{2\mathrm{d}}^{\dagger}W_{2\mathrm{d}}^{-1} \tag{4.7}$$

does not fulfill the definition of the Moore–Penrose pseudoinverse [17, Sect. 5.5.4], as $(L\widetilde{L}^{\dagger})^T \neq L\widetilde{L}^{\dagger}$ (i.e., the third condition is violated). Therefore, there is no trivial way of deriving a computationally feasible direct expression for $L^{\dagger} = (W_{2\mathrm{d}}D_{2\mathrm{d}})^{\dagger}$.

A simple remedy is to consider anyway $\widetilde{L}^{\dagger}$ (4.7) as an approximation of $L^{\dagger}$. Indeed, while $L^{\dagger}$ is characterized by

$$\begin{aligned} L^{\dagger}y &= \arg\min_{x\in\mathbb{R}^N} \|(W_{2\mathrm{d}}D_{2\mathrm{d}})x - y\|_2 = \arg\min_{x\in\mathbb{R}^N} \left\| W_{2\mathrm{d}}(D_{2\mathrm{d}}x - W_{2\mathrm{d}}^{-1}y) \right\|_2 \\ &= \arg\min_{x\in\mathbb{R}^N} \left\| D_{2\mathrm{d}}x - W_{2\mathrm{d}}^{-1}y \right\|_{W_{2\mathrm{d}}^2} , \end{aligned} \tag{4.8}$$

the matrix $\widetilde{L}^{\dagger}$ is characterized by

$$\begin{aligned} \widetilde{L}^{\dagger}y &= \arg\min_{x\in\mathbb{R}^N} \left\| D_{2\mathrm{d}}x - W_{2\mathrm{d}}^{-1}y \right\|_2 = \arg\min_{x\in\mathbb{R}^N} \left\| W_{2\mathrm{d}}^{-1}(W_{2\mathrm{d}}D_{2\mathrm{d}}x - y) \right\|_2 \\ &= \arg\min_{x\in\mathbb{R}^N} \|W_{2\mathrm{d}}D_{2\mathrm{d}}x - y\|_{W_{2\mathrm{d}}^{-2}} , \end{aligned}$$

so that $\widetilde{L}^\dagger$ can be regarded as the pseudoinverse of $W_{2\mathrm{d}} D_{2\mathrm{d}}$ computed in the $W_{2\mathrm{d}}^{-2}$ norm. Alternatively, $\widetilde{L}^\dagger$ can be regarded as the preconditioner for problem (1.1) obtained from (2.1) after the two-step transformation process

$$\begin{aligned} &\min_{x\in\mathbb{R}^N} \|Ax-b\|_2^2 + \lambda \|(W_{2\mathrm{d}} D_{2\mathrm{d}})x\|_2^2 \\ &\quad\approx \min_{z=D_{2\mathrm{d}}x} \left\| A D_{2\mathrm{d}}^\dagger z - b \right\|_2^2 + \lambda \|W_{2\mathrm{d}} z\|_2^2 \\ &\quad= \min_{y=W_{2\mathrm{d}} D_{2\mathrm{d}}x} \left\| A D_{2\mathrm{d}}^\dagger W_{2\mathrm{d}}^{-1} y - b \right\|_2^2 + \lambda \|y\|_2^2 \end{aligned} \tag{4.9}$$

has been performed. We stress that problem (4.9) is not equivalent to (2.1), as $\widetilde{L}^\dagger \neq L^\dagger$. Nevertheless, matrix-vector products with $\widetilde{L}^\dagger$ can be efficiently computed by exploiting its structure with an $O(N^{3/2})$ cost (see (4.6)). Extensive numerical experiments (some of them reported in Sect. 5) show that, in practice, $\widetilde{L}^\dagger$ (4.7) is a valid alternative to $L^\dagger$.

The other preferred approach to compute $L^\dagger$ for the 2d case (without resorting to approximations) is to employ an iterative method to solve the least-squares problem (4.8). This can be efficiently achieved by applying LSQR [26] or LSMR [13]. Both of them require a matrix-vector product with L and one with L^T, and this can be efficiently achieved with an $O(N)$ computational cost per iteration (see (4.3)). We remark that in the TV-FGMRES setting the matrix L depends on an approximation of the solution (as $W_{2\mathrm{d}} = W_{2\mathrm{d}}^{(i)} = W_{2\mathrm{d}}(x^{(i-1)})$), and since in practice the conditioning of L worsens as the vector $D_{2\mathrm{d}}x^{(i)}$ gets sparser (i.e., when an increasing number of entries of $f_\tau(D_{2\mathrm{d}}x^{(i)})$ are set to τ_2, see (2.6)), the convergence of LSQR and LSMR can be accelerated by using an appropriate preconditioner. Therefore, instead of (4.8), we consider the (right-) preconditioned least-squares problem

$$\min_{\widehat{x}\in\mathbb{R}^N} \|(W_{2\mathrm{d}} D_{2\mathrm{d}}) P_L \widehat{x} - y\|_2 \,, \quad x = P_L \widehat{x} \,. \tag{4.10}$$

In this paper we explore two choices for the preconditioner P_L that are related to the structure of the pseudoinverse of L:

$$P_L^{(1)} = D_{2\mathrm{d}}^\dagger \,, \quad \text{and} \quad P_L^{(2)} = D_{2\mathrm{d}}^\dagger W_{2\mathrm{d}}^{-1} = \widetilde{L}^\dagger. \tag{4.11}$$

We also consider a preconditioner built around the idea of approximating the row scaling $W_{2\mathrm{d}}$ by a diagonal column scaling $\widetilde{W}_{2\mathrm{d}}$ (see [22]), so that

$$D^T W_{2\mathrm{d}}^2 D \approx \widetilde{W}_{2\mathrm{d}} (D^T D) \widetilde{W}_{2\mathrm{d}} \,, \quad \text{where} \quad \mathrm{diag}(\widetilde{W}_{2\mathrm{d}}) = \sqrt{\mathrm{diag}(D^T W_{2\mathrm{d}}^2 D)./\mathrm{diag}(D^T D)} \tag{4.12}$$

(all the operations in the definition of $\widetilde{W}_{2\mathrm{d}}$ are applied component-wise). Note that $\widetilde{W}_{2\mathrm{d}}$ can be computed efficiently using the following relations

$$\mathrm{diag}(D^T W_{2\mathrm{d}}^2 D) = (D^T).^2 \mathrm{diag}(W_{2\mathrm{d}}) \,, \quad \mathrm{diag}(D^T D) = (D^T).^2 \mathbf{1} \,.$$

The third choice for the preconditioner P_L in (4.10) is the inverse of the Cholesky factor of (4.12), so that, recalling the expressions (4.4) and (4.5),

$$P_L^{(3)} = \widetilde{W}_{2\mathrm{d}}^{-1} \left[V_{1\mathrm{d}} \otimes V_{1\mathrm{d}} \right] \widetilde{D}_\beta^{-1}, \tag{4.13}$$

where $\widetilde{D}_\beta = \widetilde{D}_{1:N,1:N} + \beta I$ corresponds to taking the singular values $\{\sigma_i\}_{i=1,\ldots,N}$ of $D_{2\mathrm{d}}$, with and added tolerance $\beta > 0$ used to overcome the fact that the $D_{2\mathrm{d}}$ is rank deficient (i.e., $\sigma_N = 0$). This preconditioner is effective for a wide range of tolerances (in our numerical experiments we use $\beta = \sigma_{N-1}$).

Applying LSQR or LSMR to problem (4.10) with preconditioners (4.11) or (4.13) has an $O(N^{3/2})$ computational cost per iteration [see (4.6)]. The LSQR or LSMR iterations are terminated when the approximation x_k of the solution of (4.8) obtained at the kth iteration satisfies a stopping criterion based on the residual or the normal equations residual norm tolerance, i.e., when

$$\|y - Lx_k\|_2 / \|y\|_2 < \rho_1 \quad \text{or} \quad \|L^T(y - Lx_k)\|_2 / \|L^T y\|_2 < \rho_2\,, \quad \rho_1, \rho_2 > 0\,. \tag{4.14}$$

We remark that the quantities in (4.14) can be conveniently monitored by computing the corresponding ones for the projected problems.

An illustration of the effect of the preconditioners (4.11) and (4.13) when LSQR is used to compute $L^\dagger$ is provided in Figs. 2 and 3, where a model image deblurring problem with a small image of size 32×32 pixels is considered (analogously to Sect. 5, *Example 1*). Figure 2 displays the distribution of the (numerically) non-zero singular values of the preconditioned matrix LP_L, i.e., the first $N - 1$, for $P_L = I$ and $P_L = P_L^{(j)}$, $j = 1, 2, 3$, and clearly shows that $P_L^{(2)} = \widetilde{L}^\dagger$ is the most effective preconditioner in clustering the singular values of the preconditioned matrix LP_L around 1 and in reducing its conditioning, resulting in a fast convergence of LSQR applied to (4.10). Correspondingly, Fig. 3 displays the history of the LSQR relative errors $\|L^\dagger y - x_k\|_2 / \|L^\dagger y\|_2$ versus the number k of LSQR iterations for $P_L = I$, $P_L = P_L^{(2)}$, and $P_L = P_L^{(3)}$. In this setting, $L^\dagger y$ is computed (using the SVD of L) to get the vector z_i in line 1 of Algorithm 1, i.e., to get the solution $\bar{x}_{L,i}$ at the ith TV-FGMRES iterate, for three different values of i.

4.3 Stopping criteria

As already remarked in Sect. 1, TV-FGMRES is inherently parameter-free, as only an appropriate stopping criterion for the iterations should be chosen. All the approaches hinted in this section are mentioned in the survey paper [2], where further references and details are available. We propose to use the following strategies (adapted to the solver at hand):

– *Quasi-optimality criterion*, which prescribes to select the solution x_{L,m^*} obtained at the m^*th iteration such that

$$m^* = \arg \min_{m \le M_{\mathrm{it}}} \mathrm{TV}(x_{L,m+1} - x_{L,m})\,. \tag{4.15}$$

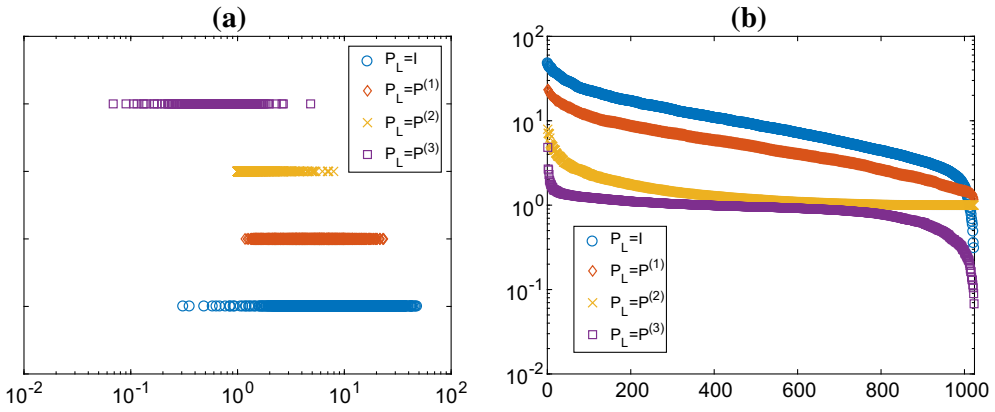


Fig. 2 Non-zero singular values of the preconditioned coefficient matrix in (4.10), with $L = (W_{2\mathrm{d}} D_{2\mathrm{d}})$ of size 1984×1024, and $P_L = I$, $P_L^{(j)}$, $j = 1, 2, 3$. **a** Distribution (clusters) of the singular values. **b** Singular values versus component number. These graphs are obtained at the 20th iteration of the TV-FGMRES method applied to the test problem in Sect. 5, *Example 1*

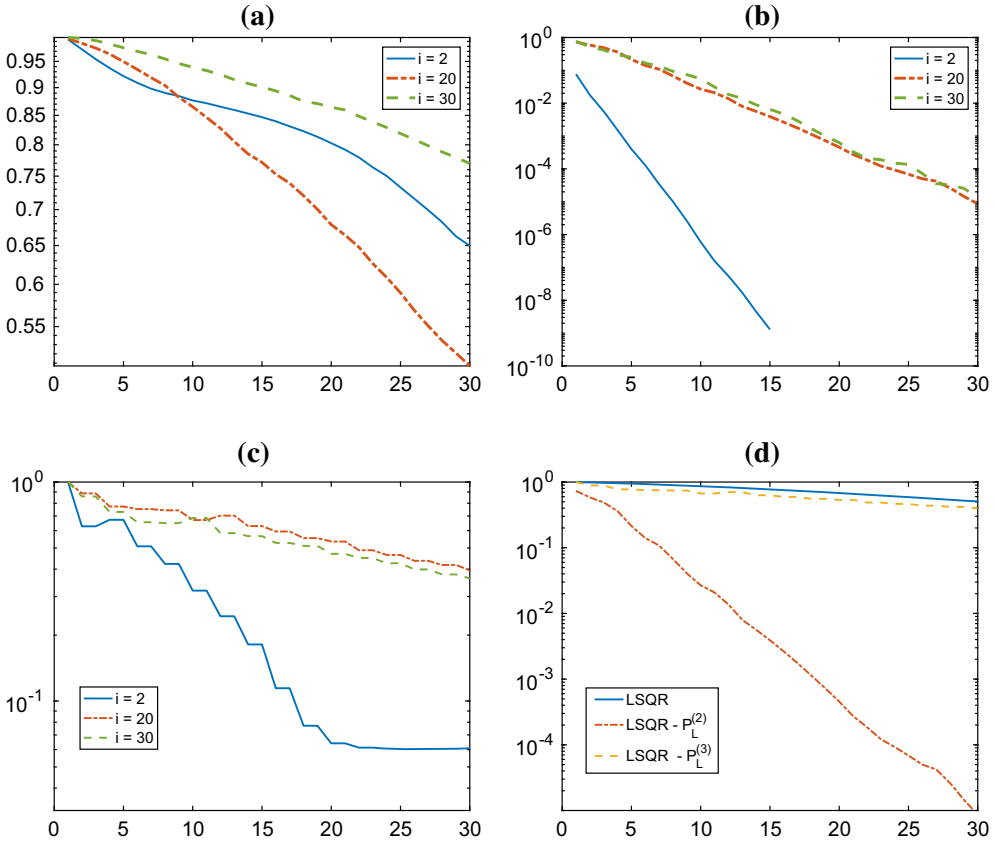


Fig. 3 History of the relative errors of the LSQR method for the computation of $L^{\dagger} y$, when employed at the ith iteration of TV-FGMRES applied to the test problem in Sect. 5, *Example 1*. **a** Without preconditioning. **b** With $P_L = P_L^{(2)}$ as preconditioner. **c** With $P_L = P_L^{(3)}$ as preconditioner. **d** Comparative history of the relative errors for LSQR with different preconditioners at the 20th iteration of TV-FGMRES (so that LSQR is applied to a least-squares problem whose coefficient matrix has the singular value distribution displayed in Fig. 2)

We remark that, although the quasi-optimality criterion requires M_{it} iterations to be performed in advance (where M_{it} is a selected maximum number of iterations), no additional computational cost per iteration has to be accounted for in order to apply (4.15) (recall the arguments at the beginning of this section).

– *Discrepancy principle*, which prescribes to stop as soon as an approximation $x_{L,m}$ is computed such that

$$\|b - Ax_{L,m}\|_2 = \|r_{L,m}\|_2 \leq \theta\epsilon\,, \tag{4.16}$$

where $\theta > 1$ is a safety threshold, and $\epsilon = \|e\|_2$ is the norm of the noise e affecting the data (1.1). The discrepancy principle is a very popular and well-established stopping criterion that relies on the availability of a good estimate of $\|e\|_2$. However, for the TV-FGMRES method, application of the discrepancy principle may significantly increase the cost per iteration, since two additional matrix-vector products with A should be performed: one to compute $\|r_{L,m}\|_2$ (which cannot be monitored in reduced dimension, as FGMRES is applied to the left-preconditioned system (3.5) or (3.6)), and one implicit in $L_A^\dagger$ (to compute $x_{L,m}$ at each iteration). For this reason, we also propose to consider the:

– *Preconditioned discrepancy principle*, which prescribes to stop as soon as an approximation $x_{L,m}$ is computed such that

$$\|\widehat{b} - \widehat{A}\bar{x}_{L,m}\|_2 = \|\widehat{r}_{m-1}\|_2 \leq \theta\widehat{\epsilon}\,, \tag{4.17}$$

where $\widehat{\epsilon}$ is the norm of the noise associated to the preconditioned problem, i.e.,

$$\begin{aligned}\widehat{\epsilon} = \|\widehat{e}\|_2 &= \|(D^\dagger)^T Pe\|_2 = \text{trace}(P^T D^\dagger (D^\dagger)^T P)\|e\|_2 \\ &= \text{trace}(P^T D^\dagger (D^\dagger)^T P)\epsilon\,.\end{aligned} \tag{4.18}$$

Although (4.17) can be monitored at no additional cost per FGMRES iteration by using projected quantities (see (3.10)), the computation of the trace in (4.18) can be prohibitive for large-scale (and possibly matrix-free) problems. We mention that, however, efficient randomized techniques can be used to handle this task (see [28]) and, most importantly, the computation of the trace should be performed only once for a system (3.5) or (3.6) of a given size, and can be done offline.

5 Numerical experiments

In this section we present three numerical test problems to investigate the performance of TV-FGMRES against other well-known approaches to total variation regularization. In each of the examples, we focus on specific aspects of the new method that we want to emphasize. Namely, the first experiment deals with a small image, which allows us to directly compute the pseudoinverse $L^\dagger$ and compare different strategies to approximate it. The second experiment is a large-scale problem (so that $L^\dagger$ cannot be computed directly) and deals with an image having low total variation. The third experiment deals with an image having higher total variation, showing the adaptability of the new

Table 1 Summary of the acronyms denoting various solvers for TV regularization, and markers denoting the various stopping criteria

Solver	Acronym	Reference	Stopping criteria	Marker
Smoothing-norm GMRES with L	GMRES(L)	[19]	(4.15), inner	Diamond
Restarted generalized AT	ReSt-GAT	[15]		
Restarted Golub–Kahan bidiag.	ReSt-GKB	[16]	(4.16), inner	Square
Fast gradient-based TV	FBTV	[3]		
TV-FGMRES for TV with $L^\dagger$	FGMRES(1)	–	(4.17), inner	Hexagon
TV-FGMRES for TV_p with $L^\dagger$	FGMRES(p)	–		
TV-FGMRES for TV_p with $\widetilde{L}^\dagger$	FGMRES($\sim p$)	–		

methods to a broader class of images. All the tests are performed running MATLAB R2017a and using some of the functionalities available within the MATLAB toolbox IR Tools [14]. Table 1 summarizes the acronyms for the various methods tested in this section, together with the markers used to denote iterations satisfying specific stopping criteria within the displayed graphs. LSQR is used to compute $L^\dagger$, possibly with the preconditioner $P = \widetilde{L}^\dagger$, allowing at most 30 iterations, and taking $\rho_1 = 10^{-8}$ for the stopping criterion in (4.14). In all the experiments, the quality of the solution is measured by the relative restoration error (RRE) $\|x_{L,m} - x^{\text{ex}}\|_2/\|x^{\text{ex}}\|_2$, where x^{ex} is the exact solution of the noise-free problem (1.1).

Example 1 We consider the task of restoring a geometrical test image of size 32×32 pixels, corrupted by a Gaussian blur with PSF analytically defined by

$$p_{i,j} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(i^2 + j^2)\right), \tag{5.1}$$

with $\sigma = 1$, $i, j = -2, -1, 0, 1, 2$, and additive Gaussian noise of relative noise level $\varepsilon_{\text{rel}} = \|e\|_2/\|b^{\text{ex}}\|_2 = 10^{-2}$, with $b^{\text{ex}} = Ax^{\text{ex}} = b - e$. The corrupted image is displayed in Fig. 5a. Since the size of this problem is moderate, it is affordable to compute directly the pseudoinverse of the matrix $L = WD$, so that we can run Algorithm 1 without resorting to (preconditioned) LSQR to perform step 1. In Fig. 4 we compare the performance of standard GMRES, GMRES(D), TV-FGMRES, TV-FGMRES for $\text{TV}_{0.1}$, a fast gradient-descent-method for TV (with a default value $\lambda = 2.9 \times 10^{-3}$), and the restarted generalized Arnoldi–Tikhonov method (with an automatically selected λ stabilizing around 4.5×10^{-2}). 50 iterations are performed by each solver. Looking at the graphs in Fig. 4a we can see that the better-performing method for this test problem is TV-FGMRES for $\text{TV}_{0.1}$. Including flexibly updated weights within TV-FGMRES clearly results in a great gain in accuracy with respect to an approach based on preconditioning GMRES with the fixed matrix $D^\dagger$. Moreover, TV-FGMRES allows to reconstruct a solution of better quality with respect to the FBTV one, with considerable computational savings. The performance of ReSt-GAT, which is still based on the Arnoldi algorithm, is not as good, since the total-variation-inspired preconditioners are not incorporated into the approximation subspace for the

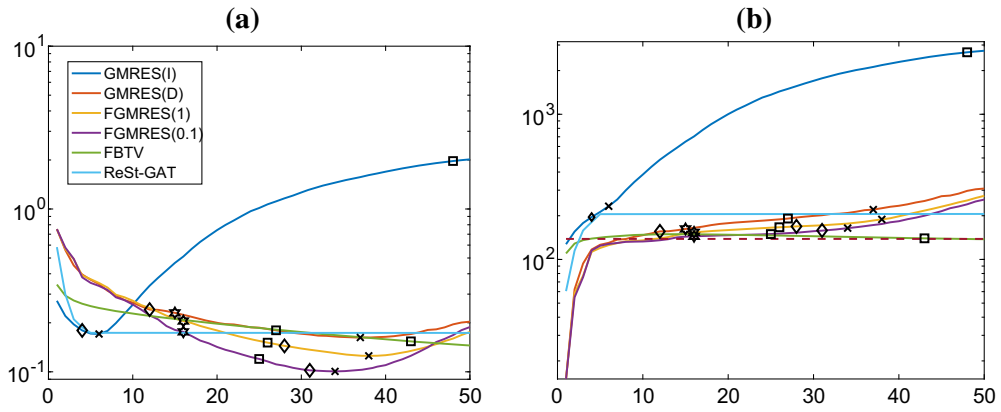


Fig. 4 *Example 1*. **a** History of the relative errors for various solvers. **b** History of the total variation of approximate solution for various solvers [line specifications as listed in frame (**a**)]; the dashed horizontal line is the total variation of the exact solution x^{ex}. The $\times$ marker highlights the iteration minimizing the relative error, while the other markers are summarized in Table 1

solution. The graphs in Fig. 4b display the value of the total variation of the approximate solution recovered at each iteration, versus the iteration number. Looking at these graphs we can clearly see that TV-FGMRES, TV-FGMRES for $\text{TV}_{0.1}$, and FBTV are the most successful methods in reconstructing approximate solutions whose total variation is the closest to the one of the exact image. The best reconstructed images for the GMRES-based methods are displayed in Fig. 5, where also surface plots thereof are provided in order to better highlight the ideally piecewise-constant features of the solutions that are fully recovered only when TV-FGMRES for $\text{TV}_{0.1}$ is employed. Relative errors for these methods are reported in the caption. Finally, in Fig. 6, we display the relative error history and total variation history obtained running different instances of the TV-FGMRES method, where the computation of the pseudoinverse $L^\dagger$ is done directly, the approximation $\widetilde{L}^\dagger = D^\dagger W^{-1}$ of $L^\dagger$ is used, or where both unpreconditioned LSQR and preconditioned (with $P = \widetilde{L}^\dagger$) LSQR (PLSQR) are employed to compute $L^\dagger$. The best attained relative errors are reported in the caption. We can clearly notice that the quantities computed by PLSQR perfectly mimic the ones obtained using $L^\dagger$ and, for this reason, in the following large-scale experiments (where computing $L^\dagger$ directly is not feasible) we confidently use PLSQR to perform this task. On the other hand, the computationally convenient approximation $\widetilde{L}^\dagger$ of $L^\dagger$ recovers a solution of lower accuracy but similar total variation. The (unpreconditioned) LSQR performance is quite remarkable in terms of the relative error, though the corresponding total variations values are not very adherent to the true ones, as a low-accuracy approximation of $L^\dagger$ is inevitably computed (recall the remarks in Sect. 4.2).

Example 2 We consider the task of restoring the well-known Shepp–Logan phantom of size 256×256 pixels, affected by a Gaussian blur whose PSF is given by (5.1), with $\sigma = 4$ and $i, j = -127, \ldots, 127$, and corrupted by Gaussian noise with relative level $\varepsilon_{\text{rel}} = 5 \times 10^{-2}$ (see Fig. 8a). In Fig. 7 we plot the values of the relative error (frame (a)) and the total variation (frame (b)) versus the number of iterations for a variety

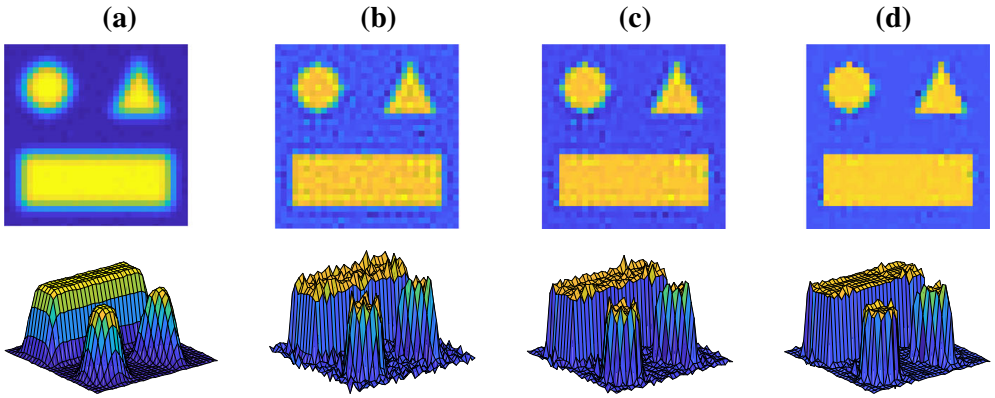


Fig. 5 *Example 1*. **a** Blurred noisy data. Best reconstructed solutions: **b** GMRES(D) (RRE, 1.6239×10^{-1}). **c** TV-FGMRES (RRE, 1.2538×10^{-1}). **d** TV-FGMRES for $\mathrm{TV}_{0.1}$ (RRE, 1.0057×10^{-1})

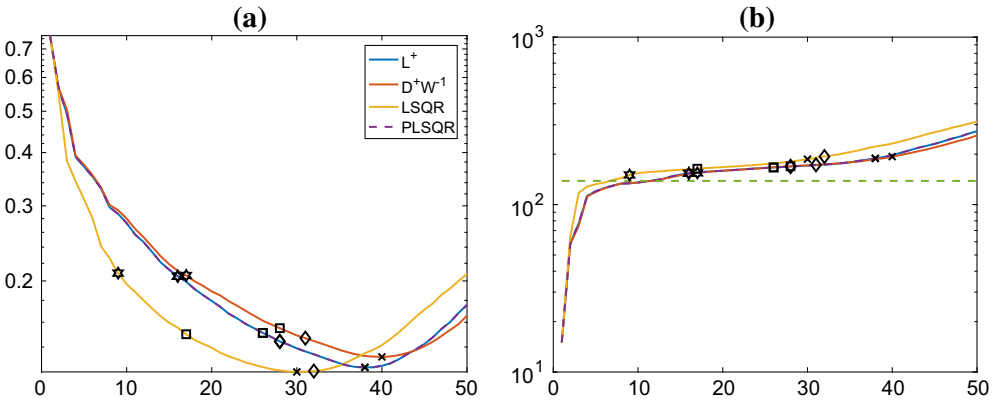


Fig. 6 *Example 1*. **a** History of the relative errors of TV-FGMRES with exact $L^\dagger$, $L^\dagger$ approximated by $\widetilde{L}^\dagger = D^\dagger W^{-1}$, $L^\dagger$ computed by LSQR, and $L^\dagger$ computed by preconditioned LSQR with $P = \widetilde{L}^\dagger$. **b** History of the total variation for TV-FGMRES (line specifications as listed in frame (**a**))

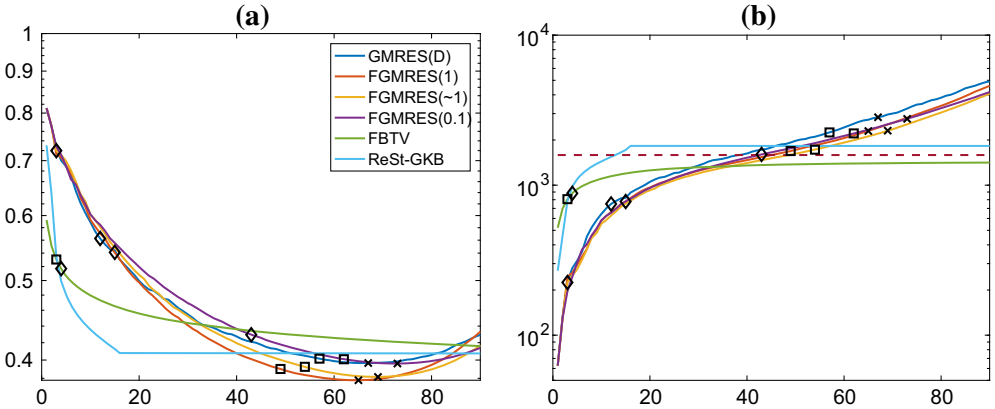


Fig. 7 *Example 2*. **a** History of the relative errors for various solvers. **b** History of the total variation of approximate solution for various solvers (line specifications as listed in frame (**a**)); the dashed horizontal line is the total variation of the exact solution x^{ex}. The $\times$ marker highlights the iteration minimizing the relative error, while the other markers are summarized in Table 1

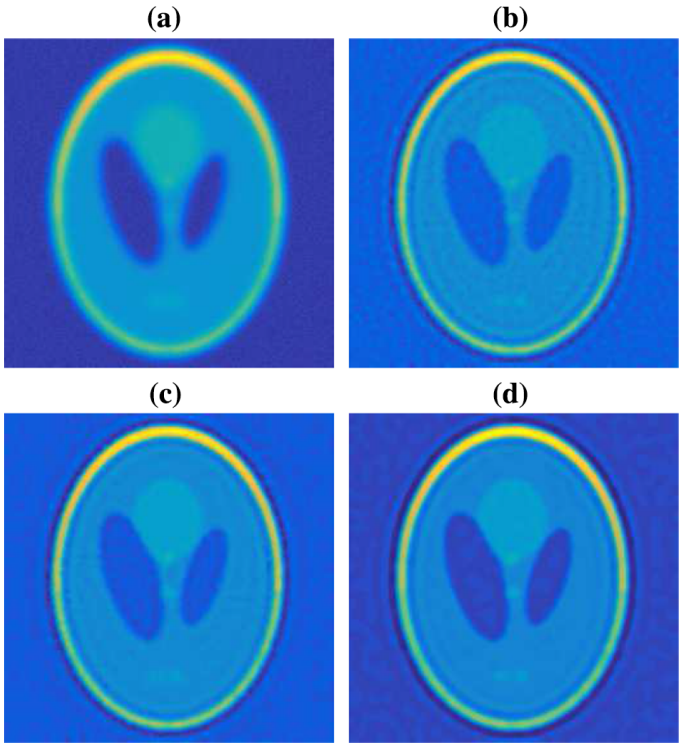


Fig. 8 *Example 2*. **a** Blurred noisy data. Restored solutions when the discrepancy principle is satisfied: **b** GMRES(D) (RRE, 4.0162×10^{-1}; it, 57). **c** TV-FGMRES (RRE, 3.9013×10^{-1}; it, 49). **d** fast gradient-based method for TV (RRE, 4.1600×10^{-1}; it, 90)

of solvers for (1.3): the layout of this figure is similar to the one of Fig. 4, and 90 iterations are performed for each solver. We can clearly see that, for this test problem, TV-FGMRES is the most effective solver, which attains better accuracy in the least number of iterations. The fast gradient-based method for TV (with a default value $\lambda = 5.4 \times 10^{-4}$) seems quite slow for this problem, and the restarted GKB algorithm (which is basically the restarted GAT method, where Golub–Kahan bidiagonalization is considered instead of the Arnoldi algorithm) rapidly stagnates (with an automatically selected λ stabilizing around 1.7×10^{-2}).

Figure 8 displays the phantoms restored when the discrepancy principle (4.16) is satisfied by the GMRES(D), the TV-FGMRES, and the FBTV methods (the latter does not stop within the maximum number of allowed iterations). Relative errors and corresponding iteration numbers are reported in the caption. We can clearly see that the TV-FGMRES solution is the one with lower relative reconstruction error, though the FBTV solution surely appears more blocky (containing also some artifacts). On the opposite, the GMRES(D) solution displays many ringing artifacts, which are partially removed when adaptive weights are incorporated within the TV-FGMRES preconditioners and approximation subspace.

Example 3 We consider the task of restoring the cameraman test image of size 256×256 pixels, corrupted by the same blur and noise used with the previous example (see

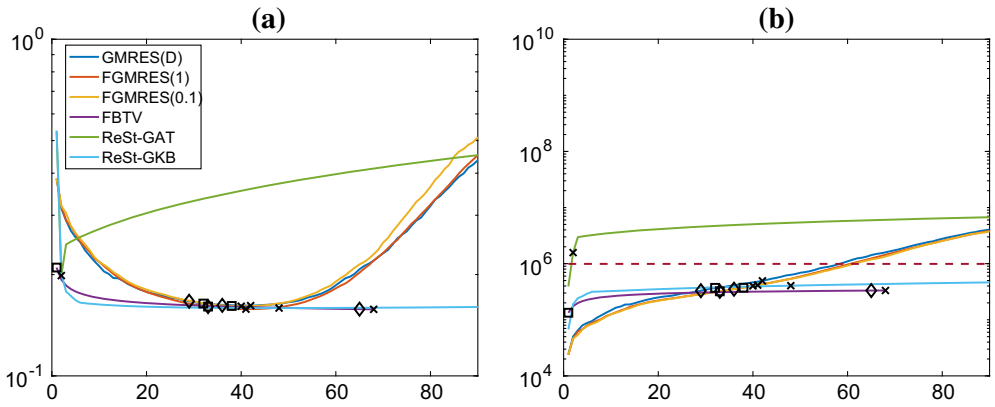


Fig. 9 *Example 3*. **a** History of the relative errors for various solvers. **b** History of the total variation of approximate solution for various solvers (line specifications as listed in frame (**a**)); the dashed horizontal line is the total variation of the exact solution x^{ex}. The $\times$ marker highlights the iteration minimizing the relative error, while the other markers are summarized in Table 1

Fig. 10a). However, contrarily to the previous example, the total variation of the exact image is quite moderate. In Fig. 9 we plot the values of the relative error (frame (a)) and the total variation (frame (b)) versus the number of iterations for a variety of solvers for (1.3): the layout of this figure is similar to the one of Figs. 4 and 7. Also for this example, 90 iterations are performed for each solver. The best reconstructions computed by the GMRES(D), the TV-FGMRES, and the FBTV methods are displayed in Fig. 10 (relative restoration errors are reported in the caption). For this test problem all the solvers seem to have a similar performance in terms of relative errors (except for ReSt-GAT that exhibits an unstable behavior because of a likely inappropriate choice of the regularization parameter). We also remark that both ReSt-GKB and FBTV are very fast in recovering an approximate solution, whose quality however stagnates. TV-FGMRES seems to recover a more accurate value of the total variation of the approximate solutions along the iterations. Correspondingly, more details are visible in the image restored by TV-FGMRES with respect to the one restored by the FBTV method, which is more blocky (coherently to the fact that FBTV underestimates the total variation of the exact solution).

6 Conclusion and future work

In this paper we presented a novel GMRES-based approach for computing regularized solutions for large-scale linear inverse problems involving TV penalization, with applications to image deblurring problems. By considering an IRN approach to approximate the non-differentiable total variation term, and by exploiting the framework of smoothing-norm preconditioning for GMRES, we could derive the TV-FGMRES method that leverages the flexible Arnoldi algorithm. The TV-FGMRES method easily extends to problems involving TV_p regularization, and it is inherently parameter-free and efficient, as various numerical experiments and comparisons with other solvers for total variation regularization show.

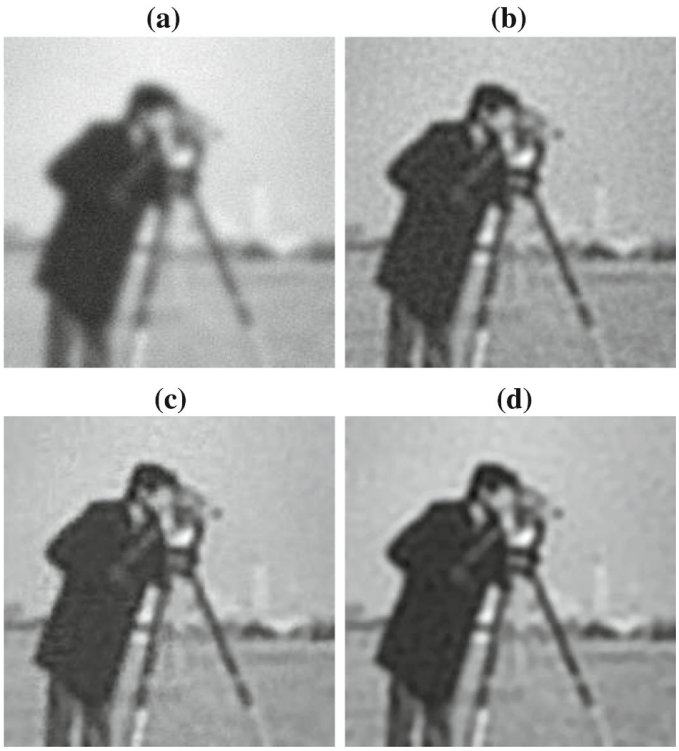


Fig. 10 *Example 3*. **a** Blurred noisy image. Best restored solutions obtained by: **b** GMRES(D) (RRE, 1.6170×10^{-1}). **c** TV-FGMRES (RRE, 1.5793×10^{-1}). **d** fast gradient-based method for TV (RRE, 1.5931×10^{-1})

Future work includes a more careful investigation of how to optimally derive alternative preconditioners that can speed-up the convergence of LSQR for the computation of the pseudo-inverse $L^{\dagger}$ for large-scale problems. Strategies to extend the TV-FGMRES method to incorporate additional penalization terms can be studied as well. Finally, ways of extending TV-FGMRES to handle non-square coefficient matrices can be devised, by exploiting the flexible Golub–Kahan bidiagonalization algorithm derived in [10].

Acknowledgements We are grateful to the anonymous Referees for providing insightful suggestions that helped to improve the paper. We would also like to thank James Nagy for insightful discussions about structured matrix computations.

References

1. Arridge, S.R., Betcke, M.M., Harhanen, L.: Iterated preconditioned LSQR method for inverse problems on unstructured grids. Inverse Probl. **30**(7), 075009 (2014)
2. Bauer, F., Gutting, M., Lukas, M.A.: Evaluation of Parameter Choice Methods for Regularization of Ill-Posed Problems in Geomathematics, pp. 1713–1774. Springer, Berlin (2015)

3. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans. Image Process. **18**(11), 2419–2434 (2009)
4. Berisha, S., Nagy, J.G.: Iterative image restoration. In: Chellappa, R., Theodoridis, S. (eds.) Academic Press Library in Signal Processing, chap. 7, vol. 4, pp. 193–243. Elsevier, Amsterdam (2014)
5. Calvetti, D.: Preconditioned iterative methods for linear discrete ill-posed problems from a Bayesian inversion perspective. J. Comput. Appl. Math. **198**(2), 378–395 (2007)
6. Calvetti, D., Lewis, B., Reichel, L.: On the regularizing properties of the GMRES method. Numer. Math. **91**(4), 605–625 (2002)
7. Candés, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted l1 minimization. J. Fourier Anal. Appl. **14**, 877–905 (2008)
8. Chan, T.F., Golub, G.H., Mulet, P.: A nonlinear primal-dual method for total variation-based image restoration. SIAM J. Sci. Comput. **20**(6), 1964–1977 (1999)
9. Chan, T.F., Shen, J.: Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods. SIAM, Philadelphia (2005)
10. Chung, J., Gazzola, S.: Flexible Krylov methods for ℓ^p regularization (2018) (**submitted**)
11. Chung, J., Knepper, S., Nagy, J.G.: Large-scale inverse problems in imaging. In: Scherzer, O. (ed.) Handbook of Mathematical Methods in Imaging, chap. 2, pp. 43–86. Springer, Berlin (2011)
12. Eldén, L.: A weighted pseudoinverse, generalized singular values, and constrained least-squares problems. BIT Numer. Math. **22**(4), 487–502 (1982)
13. Fong, D.C.L., Saunders, M.A.: LSMR: an iterative algorithm for sparse least-squares problems. SIAM J. Sci. Comput. **33**(5), 2950–2971 (2011)
14. Gazzola, S., Hansen, P.C., Nagy, J.G.: IR tools: a MATLAB package of iterative regularization methods and large-scale test problems. Numer. Algorithms (2018). https://doi.org/10.1007/s11075-018-0570-7
15. Gazzola, S., Nagy, J.G.: Generalized Arnoldi–Tikhonov method for sparse reconstruction. SIAM J. Sci. Comput. **36**(2), B225–B247 (2014)
16. Gazzola, S., Novati, P., Russo, M.R.: On Krylov projection methods and Tikhonov regularization. Electron. Trans. Numer. Anal. **44**(1), 83–123 (2015)
17. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins, Baltimore (1996)
18. Hansen, P.C.: Discrete Inverse Problems: Insight and Algorithms. Society for Industrial and Applied Mathematics, Philadelphia (2010)
19. Hansen, P.C., Jensen, T.K.: Smoothing-norm preconditioning for regularizing minimum-residual methods. SIAM J. Matrix Anal. Appl. **29**(1), 1–14 (2007)
20. Hansen, P.C., Nagy, J.G., O'Leary, D.P.: Deblurring Images: Matrices, Spectra, and Filtering. Society for Industrial and Applied Mathematics, Philadelphia (2006)
21. Jensen, T.K., Hansen, P.C.: Iterative regularization with minimum-residual methods. BIT Numer. Math. **47**, 103–120 (2007)
22. Kubínová, M., Nagy, J.G.: Robust regression for mixed Poisson–Gaussian model. Numer. Algorithms **79**(3), 825–851 (2018)
23. Lanza, A., Morigi, S., Reichel, L., Sgallari, F.: A generalized Krylov subspace method for $\ell_p - \ell_q$ minimization. SIAM J. Sci. Comput. **37**, S30–S50 (2015)
24. Notay, Y.: Flexible conjugate gradients. SIAM J. Sci. Comput. **22**, 1444–1460 (2000)
25. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. Multiscale Model. Simul. **4**(2), 460–489 (2005)
26. Paige, C.C., Saunders, M.A.: LSQR: an algorithm for sparse linear equations and and sparse least squares. ACM Trans. Math. Softw. **8**(1), 43–71 (1982)
27. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2003)
28. Saibaba, A.K., Alexanderian, A., Ipsen, I.C.F.: Randomized matrix-free trace and log-determinant estimators. Numer. Math. **137**(5), 353–395 (2017)
29. Vogel, C.R., Oman, M.E.: Fast, robust total variation-based reconstruction of noisy, blurred images. IEEE Trans. Image Process. **7**(6), 813–824 (1998)
30. Wohlberg, B., Rodríguez, P.: An iteratively reweighted norm algorithm for minimization of total variation functionals. IEEE Signal Process. Lett. **14**, 948–951 (2007)

4.3 Conclusions

This chapter presents yet another case where (flexible) Krylov methods can be efficiently used in combination with variational regularization schemes. In particular, it presents a new algorithm, TV-FGMRES, to approximate total variation regularization. The principles behind the algorithmic framework used in this chapter are conceptually very similar to the ones in Chapter 3, and possible extensions of the methods presented in Chapter 4 and 3 to other variational regularization schemes may be considered in the future. However, as seen in [GL19] for TV regularization, higher complexity of the regularization term requires more cumbersome derivations which need more sophisticated implementation strategies.

Chapter 5

Conclusions and outlook

This thesis presents three pieces of original research in the area of regularization techniques for large-scale linear discrete ill-posed problems. These include a new principled algorithmic framework for Krylov-Tikhonov methods that automatically sets the regularization parameter, and new algorithms for ℓ_2-ℓ_p and total variation regularization.

Krylov methods and Tikhonov regularization are a very powerful combination to solve large-scale linear ill-posed problems, as shown by their extensive use in different applications. This area of research has recently regained attention with the development of new, modified and generalized Krylov subspace methods. In particular, as presented in Chapters 3 and 4, flexible Krylov methods can be used to find solutions to complex classical regularization schemes by constructing a sequence of suitable quadratic approximations thereof. This general approach, that has been shown to be very competitive, opens the door to the extension to other classical regularization schemes. Just as an example, a natural extension of these methods could be derived for group sparsity regularization [CHLH18, CHHL14]. Another straightforward extension of the work in Chapter 4 is the generalization of the algorithm presented in [GL19] to non-square systems. This could be based on the flexible Golub-Kahan algorithm presented in [CG19] (already used in Chapter 3), and could be applied to more problems such as computed tomography (CT).

Another research area that is closely related to the content of this thesis is the theoretical analysis of flexible Krylov methods. First, Chapter 3 provides a convergence proof for a specific application of flexible Krylov methods used in combination with an iteratively reweighted scheme, as well as a re-formulation of flexible Krylov methods as particular instances of augmented Krylov subspace methods. However, proving more general statements about convergence of flexible Krylov methods and their regularization properties is still a topic of ongoing research. Second, there is still room for a deeper theoretical understanding on how the prior information coming from suitable prior-conditioning is embedded in the flexible Krylov subspaces. From my point of view, a very interesting insight can be found in Chapter 4, where the direct effect of edge-preserving regularization can be observed in the basis vectors for the search space of a one-dimensional deconvolution example [GL19, Section 3].

Finally, a better understanding of flexible Krylov methods can be used to develop efficient recycling techniques for flexible Krylov subspaces. When employing flexible Krylov methods we usually deal with iteration-dependent preconditioners that improve over the iterations. This is because, for many flexible solvers, the preconditioners depend on the most recent available approximation of the solution, which improves as the iterations proceed. In particular, the first basis vectors incorporated in the solution subspace, which

for the purposes of regularization are often the most significative ones, are also associated to the least updated (and therefore "worst") preconditioners. For this reason, restarting, and therefore building a new Krylov subspace for the solution that is more tailored to the specific problem we are solving, could lead in theory to better reconstructions for the solution, as well as to limit the amount of required memory to solve the problem.

In the framework of inverse problems, where the preconditioning is associated to prior information on the solution, special attention should be put in the restarting strategy. While classic restarting proposes constructing a Krylov subspace for the correction of the solution after the restart, this might be inappropriate for regularization if the information encoded in the preconditioners is just related to the properties of the solutions and not the corrections. In combination with restarting, a common strategy to reduce computational cost of Krylov methods is to use recycling strategies to select a "good" set of vectors from the previous Krylov subspace that are incorporated in the search space after the restart. Traditional recycling techniques, motivated by a slowly changing sequence of linear systems or limited computational memory, are built to preserve a subspace that significatively contributes to reducing the residual norm. In the framework of ill-posed problems, where the solution subspace should promote some desirable features of the regularized solution, different strategies should be applied to choose what is a "good" set of vectors to carry forward through the iterations. Subspace recycling for iterative methods is a popular topic, see for example [SdSK20], and, to the best of our knowledge, no work concerning flexible Krylov subspaces has been published yet.

Concluding, Krylov methods are well-known and renowned iterative methods for large-scale linear ill-posed problems, and are still a main topic of research. The work presented in this thesis aims to make a contribution in this area, and it will hopefully convince the reader that flexible Krylov subspaces are both adaptable and reliable, and can be exploited much further to deal with complex regularization schemes, while maintaining attractive theoretical guarantees.

Bibliography

[Bar13] J.L. Barlow. Reorthogonalization for the Golub–Kahan–Lanczos bidiagonal reduction. *Numer. Math.*, 124:237–278, 2013.

[BN06] J. M. Bardsley and J. G. Nagy. Covariance-preconditioned iterative methods for nonnegatively constrained astronomical imaging. *SIAM J. Matrix Anal. Appl.*, 27(4):1184–1197, 2006.

[BT09] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[Cal07] D. Calvetti. Preconditioned iterative methods for linear discrete ill-posed problems from a Bayesian inversion perspective. *J. Comput. Appl. Math.*, 198(2):378–395, 2007. Special Issue: Applied Computational Inverse Problems.

[CG19] J. Chung and S. Gazzola. Flexible Krylov methods for ℓ_p regularization. *SIAM J. Sci. Comput.*, 41(5):S149–S171, 2019.

[CHHL14] C. Chen, J. Huang, L. He, and H. Li. Preconditioning for accelerated iteratively reweighted least squares in structured sparsity reconstruction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2713–2720, 2014.

[CHLH18] C. Chen, L. He, H. Li, and J. Huang. Fast iteratively reweighted least squares algorithms for analysis-based sparse reconstruction. *Medical Image Analysis*, 49:141–152, 2018.

[CKO15] J. M. Chung, M. E. Kilmer, and D. P. O'Leary. A framework for regularization via operator approximation. *SIAM J. Sci. Comput.*, 37(2):B332–B359, 2015.

[CNO08] J. Chung, J. G. Nagy, and D. P. O'Leary. A weighted GCV method for Lanczos hybrid regularization. *Electron. Trans. Numer. Anal.*, 28:149–167, 2008.

[CS07] D. Calvetti and E. Somersalo. *Introduction to Bayesian Scientific Computing.* Springer-Verlag New York, 2007.

[DDFG10] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Comm. Pure Appl. Math.*, 63(1):1–38, 2010.

[DEMSC06] M. Donatelli, C. Estatico, A. Martinelli, and S. Serra-Capizzano. Improved image deblurring with anti-reflective boundary conditions and re-blurring. *Inverse Problems*, 22(6):2035–2053, 2006.

[EHN96] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Mathematics and Its Applications. Springer Netherlands, 1996.

[Eld82] L. Eldén. A weighted pseudoinverse, generalized singular values, and constrained least squares problems. *BIT*, 22(4):487–502, 1982.

[FR11] M. Fornasier and H. Rauhut. *Compressive Sensing*, pages 187–228. Springer New York, 2011.

[FR13] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013.

[GHN18] S. Gazzola, P. C. Hansen, and J. G. Nagy. IR tools: a MATLAB package of iterative regularization methods and large-scale test problems. *Numer. Algorithms*, 2018.

[GK65] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *J. Soc. Indust. Appl. Math. Ser. B Numer. Anal.*, 2(2):205–224, 1965.

[GL96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, third edition, 1996.

[GL19] S. Gazzola and M. Sabaté Landman. Flexible GMRES for total variation regularization. *BIT*, 59:721–746, 2019.

[GL20] S. Gazzola and M. Sabaté Landman. Krylov methods for inverse problems: Surveying classical, and introducing new, algorithmic approaches. *GAMM-Mitteilungen*, page e202000017, 2020.

[GN14] S. Gazzola and P. Novati. Automatic parameter setting for Arnoldi-Tikhonov methods. *J. Comput. Appl. Math.*, 256:180–195, 2014.

[GNL21] S. Gazzola, J.G. Nagy, and M. Sabaté Landman. Iteratively reweighted FGMRES and FLSQR for sparse reconstruction. *SIAM J. Sci. Comput.*, In press, 2021.

[GVL96] G. H. Golub and C. F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, 1996.

[Had52] J. Hadamard. Lectures on Cauchy's problem in linear partial differential equations. Dover Publications V, 316, New York., 1952.

[Han90] P. C. Hansen. The discrete Picard condition for discrete ill-posed problems. *BIT*, 30(4):658–672, 1990.

[Han98] P. C. Hansen. *Rank-deficient and discrete ill-posed problems*. SIAM, Philadelphia, 1998.

[Han10] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. SIAM, Philadelphia, 2010.

[HH93] M. Hanke and P. C. Hansen. Regularization methods for large-scale problems. *Surv. Math. Ind.*, 3:253–315, 1993.

[HJ07] P. C. Hansen and T. K. Jensen. Smoothing-norm preconditioning for regularizing minimum-residual methods. *SIAM J. Matrix Anal. Appl.*, 29(1):1–14, 2007.

[HJ17] Y. Huang and Z. Jia. Some results on the regularization of LSQR for large-scale discrete ill-posed problems. *Sci. China Math.*, 60:701–718, 2017.

[HLM+17] G. Huang, A. Lanza, S. Morigi, L. Reichel, and F. Sgallari. Majorization-minimization generalized Krylov subspace methods for l_p-l_q optimization applied to image restoration. *BIT*, 57(2):351–378, Jun 2017.

[HNO06] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering.* SIAM, Philadelphia, 2006.

[Hof86] B. Hofman. *Regularization for Applied Inverse and Ill-Posed Problems.* Teubner, Stuttgart, 1986.

[Mat] *Image Processing Toolbox Release 2020b.* The MathWorks, Inc., Natick, Massachusetts, United States.

[NPP] J. G. Nagy, K.M. Palmer, and L. Perrone. Iterative methods for image deblurring: a matlab object oriented approach. *Numer. Algorithms, 36 (2004),pp.73-93.* See also `http://www.mathcs.emory.edu/~nagy/RestoreTools`.

[OBG+05] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.*, 4(2):460–489, 2005.

[OS81] D. P. O'Leary and J. A. Simmons. A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. *SIAM J. Sci. Statist. Comput.*, 2(4):474–489, 1981.

[RHM10] R. A. Renaut, I. Hnětynková, and J. Mead. Regularization parameter estimation for large-scale Tikhonov regularization using a priori information. *Computational Statistics & Data Analysis*, 54(12):3430 – 3445, 2010.

[RR13] L. Reichel and G. Rodriguez. Old and new parameter choice rules for discrete ill-posed problems. *Numer. Algorithms*, 63:65–87, 2013.

[RS08] L. Reichel and A. Shyshkov. A new zero-finder for Tikhonov regularization. *BIT*, 48:627–643, 2008.

[RVEA17] R. Renaut, S. Vatankhah, and V. E. Ardestani. Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV. *SIAM J. Sci. Comput.*, 39:B221–B243, 2017.

[Saa03] Y. Saad. *Iterative Methods for Sparse Linear Systems.* SIAM, Philadelphia, 2nd edition, 2003.

[SdSK20] K. Soodhalter, E. de Sturler, and M. Kilmer. A survey of subspace recycling iterative methods. *GAMM-Mitteilungen*, 43, 09 2020.

[SGG+08] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational Methods in Imaging.* Springer, New York., 1 edition, 2008.

[Sil] S. Siltanen. Examples of Inverse Problems. `http://www.siltanen-research.net/IPexamples/xray_tomography/tomographic_imaging_with_sparse_data`. Accessed: 2017-09-20.

[SS86] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual method for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 7:856–869, 1986.

[VO98] C. R. Vogel and M. E. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Trans. Image Process.*, 7(6):813–824, 1998.

[Vog02] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, 2002.

[WNF09] S. J. Wright, R. D. Nowak, and M.A.T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 57(7):2479–2493, 2009.

[WR08a] B. Wohlberg and P. Rodriguez. An efficient algorithm for sparse representations with ℓ^p data fidelity term. *In Proceedings of the 4th IEEE Andean Technical Conference (AN-DESCON)*, 2008.

[WR08b] B. Wohlberg and P. Rodriguez. An iteratively reweighted norm algorithm for minimization of total variation functionals. *Signal Processing Letters, IEEE*, 14:948–951, 01 2008.